

UNIVERSIDAD CARLOS III DE MADRID



INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

PROYECTO FIN DE CARRERA

ANÁLISIS, DISEÑO Y DESARROLLO DE UN PORTAL WEB COMERCIAL

Autor: Jesús Alajarín Perera

Tutores: Luis Miguel Sanchez Garcia
Maria Cristina Marinescu



ÍNDICE DEL CONTENIDO

1.- INTRODUCCIÓN	9
1.1.- Contexto.....	10
1.2.- Motivación.....	14
1.3.- Definición del problema	17
1.4.- Estructura del documento	19
1.4.1.- Introducción.....	19
1.4.2.- Estructura del documento	19
2.- ANÁLISIS.....	21
2.1.- Tecnologías web	22
2.1.1.- Introducción.....	22
2.1.2.- Comparativa de Tecnologías web.....	23
2.1.3.- Tecnologías web seleccionadas	26
2.1.4.- Empleo de tecnologías web	33
2.2.- Ejemplos reales.....	35
2.2.1.- Introducción.....	35
2.2.2.- Ejemplos	35
2.3.- Requisitos	38
2.3.1.- Introducción.....	38
2.3.2.- Requisitos del sistema:	40
2.4.- Casos de uso	49
2.4.1.- Introducción.....	49
2.4.2.- Tipos de usuario que interactúan con la aplicación	49
2.4.3.- Casos de uso	50
2.4.4.- Especificación textual de los casos de uso	53
2.5.- Características de los usuarios	58
2.5.1.- Introducción.....	58
2.5.2.- Restricciones.....	58
2.5.3.- Suposiciones y Dependencias.....	59
2.5.4.- Requisitos Futuros	60
2.5.5.- Requisitos Específicos.....	60
2.5.6.- Requisitos de Interfaces Externas.....	62
2.5.7.- Requisitos de Rendimiento	62
2.5.8.- Restricciones del Diseño	62
2.6.- Arquitectura de la base de datos	63
2.6.1.- Introducción.....	63
2.6.2.- Arquitectura de la base de datos	63
2.7.- Planificación	67
2.7.1.- Introducción.....	67
2.7.2.- Estimación	69
2.7.3.- Estimación de los Puntos de Función	69
2.7.4.- Estimación de las líneas de código	74
2.7.5.- Estimación del esfuerzo, duración del proyecto y personal necesario.....	74
2.7.6.- Estimación del tiempo y coste del proyecto	80
2.7.7.- Estimación del tiempo y coste del proyecto por fases.	82
2.8.- Gráfico Gantt	83
2.8.1.- Introducción.....	83
2.8.2.- Diagrama Gantt	84



3.- DISEÑO	86
3.1.- Introducción	87
3.2.- Disposición de las páginas	89
3.2.1.- Introducción	89
3.2.2.- Inicio de la página principal	90
3.2.3.- Sección estática	91
3.2.4.- Sección dinámica	92
3.2.5.- Pagina de inicio del gestor	93
3.2.6.- Página de gestión de las secciones	94
3.2.7.- Pagina de creación y modificación de elementos de una sección	95
3.3.- Diseño de la base de datos	96
3.3.1.- Introducción	96
3.3.2.- Diagrama Entidad/Relación	96
3.3.3.- Determinación de las entidades	97
3.3.4.- Determinación de las relaciones existentes entre las entidades	98
3.3.5.- Supuestos semánticos no reflejados	98
3.3.6.- Modelo Relacional	99
3.3.7.- Diagrama de clases	100
4.- DESARROLLO	101
4.1.- Introducción	102
4.2.- Mapa web	104
4.2.1.- Introducción	104
4.2.2.- Mapa web	104
4.3.- Capa de presentación	105
4.3.1.- Introducción	105
4.3.2.- Página principal	106
4.3.2.1.- Menú	106
4.3.2.2.- Sección Banner	108
4.3.2.3.- Sección Smart Food	110
4.3.2.4.- Sección Siempre fresco!	111
4.3.2.5.- Sección Sushi	112
4.3.2.7.- Noticias	116
4.3.2.8.- Contacto	118
4.3.2.9.- Franquicia	120
4.3.3.- Gestor	121
4.3.3.1.- Gestor Restaurantes	125
4.3.3.2.- Nuevo Restaurante	126
4.3.3.3.- Editar Restaurante	129
4.3.3.4.- Gestor Noticias	131
4.3.3.5.- Nueva Noticia	133
4.3.3.6.- Editar Noticia	135
4.3.3.7.- Gestor Banner	137
4.3.3.8.- Nueva diapositiva	138
4.3.3.9.- Editar diapositiva	140
4.4.- Capa de Negocio	141
4.4.1.- Introducción	141
4.4.2.- Página principal	142
4.4.2.1.- Sección Banner	143
4.4.2.2.- Sección Restaurantes	144
4.4.2.3.- Sección Noticias	146
4.4.2.4.- Sección Contacto	149
4.4.3.- Gestor	150



4.4.3.1- Gestor Restaurantes	152
4.4.3.2- Nuevo Restaurante	154
4.4.3.3- Editar Restaurante	160
4.4.3.4- Gestor Noticias	165
4.4.3.5 - Nueva Noticia	168
4.4.3.6 - Editar Noticia	172
4.4.3.7 – Gestor Banner	176
4.4.3.8 – Nueva dispositiva	179
4.4.3.9 – Editar diapositiva	181
4.5.- Capa de Datos	184
5.- BATERIA DE PRUEBAS	185
5.1.- Introducción	186
5.2.- Visualización	186
5.3.- Funcionamiento	187
6.- CONCLUSIONES	190
6.1.- Conclusiones	191
6.2.- Trabajos futuros	192
7.- ANEXOS.....	194
Anexo I: Presupuesto.	195
Anexo II: Instalación.....	197
Anexo III: Manual de usuario	198
8.- BIBLIOGRAFÍA	208
8.1.- Bibliografía web	209
8.2.- Bibliografía	210

ÍNDICE DE FIGURAS

FIGURA 1 Hipertexto.....	11
FIGURA 2 NeXTcube usado por Berners-Leeen. El CERN Primer servidor web.....	12
FIGURA 3 Logo JavaScript.....	28
FIGURA 4 Logo PHP.....	30
FIGURA 5 Logo MySQL.....	31
FIGURA 6 Logo jQuery.....	32
FIGURA 7 Estructura del portal web comercial.....	34
FIGURA 8 Página web de Kia Picanto.....	35
FIGURA 9 Página web de gudanghome.....	36
FIGURA 10 Página web de Amazon.....	37
FIGURA 11 Casos de uso parte 1.....	50
FIGURA 12 Casos de uso parte 2.....	51
FIGURA 13 Dependencias.....	59
FIGURA 14 Componentes lógicos de acceso a datos.....	65
FIGURA 15 Ciclo de vida “Entrega Evolutiva”.....	68
FIGURA 16 Factores de costo Modelo Post-Arquitectura. [Boehm 1995/1] [Boehm 1995/2]..	76
FIGURA 17 Factor Multiplicador CLPX. Complejidad del producto.....	77
FIGURA 18 COCOMO II.....	81
FIGURA 19 Factores de costo Modelo Post-Arquitectura en COCOMO II.....	81
FIGURA 20 Diagrama Gantt parte 1.....	84
FIGURA 21 Diagrama Gantt parte 2.....	85
FIGURA 22 Diseño de tipo “Inicio de la página principal”.....	90
FIGURA 23 Diseño de tipo “Sección estática”.....	91
FIGURA 24 Diseño de tipo “Sección dinámica”.....	92
FIGURA 25 Diseño de tipo “Inicio gestor”.....	93
FIGURA 26 Diseño de tipo “Página de gestión de las secciones”.....	94
FIGURA 27 Diseño de tipo “Página de creación y modificación”.....	95
FIGURA 28 Diagrama entidad/relación.....	96
FIGURA 29 Modelo relacional.....	99
FIGURA 30 Diagrama de clases.....	100
FIGURA 31 Definición gráfica de la programación por capas.....	102
FIGURA 32 Esquema de la programación por capas.....	103
FIGURA 33 Menú web.....	106
FIGURA 34 Diapositiva del Banner con el texto a la izquierda.....	108
FIGURA 35 Diapositiva del Banner con el texto a la derecha.....	109
FIGURA 36 Sección SmartFood.....	110
FIGURA 37 Sección Siempre fresco!.....	111
FIGURA 38 Sección Sushi.....	112
FIGURA 39 Menú restaurantes abiertos.....	113
FIGURA 40 Menú restaurantes próximas aperturas.....	114
FIGURA 41 Noticia con foto.....	116
FIGURA 42 Noticia sin foto.....	117
FIGURA 43 Sección contacto.....	118
FIGURA 44 Aviso legal.....	119
FIGURA 45 Sección Franquicia.....	120
FIGURA 46 Página de inicio del gestor.....	121
FIGURA 47 Editor de texto Nice Edit.....	123
FIGURA 48 Calendario.....	124
FIGURA 49 Gestor de restaurantes.....	125



FIGURA 50 Gestor: Nuevo restaurante	126
FIGURA 51 Gestor: Editar restaurante	129
FIGURA 52 Gestor noticias	131
FIGURA 53 Gestor noticias cambio de la noticia destacada	132
FIGURA 54 Gestor: Nueva noticia	134
FIGURA 55 Gestor: Editar noticia	136
FIGURA 56 Gestor Banner	137
FIGURA 57 Gestor: Nueva diapositiva para el banner	138
FIGURA 58 Gestor: Editar diapositiva para el banner	140
FIGURA 59 Explicación menú y banner	199
FIGURA 60 Explicación sección restaurantes	200
FIGURA 61 Explicación sección noticias	201
FIGURA 62 Explicación sección restaurantes del gestor	203
FIGURA 63 Explicación sección noticias del gestor	205
FIGURA 64 Explicación sección banner del gestor	206

ÍNDICE DE TABLAS

Tabla 1. Comparación de tecnologías web.....	23
Tabla 2. Comparación de bases de datos SQL	25
Tabla 3. R1: Web impactante	40
Tabla 4. R2: Fácil de navegar.....	40
Tabla 5. R3: Tipografía de la letra legible y con el estilo de la web	40
Tabla 6. R4: Diseño atractivo.....	41
Tabla 7. R5: Contenido fácil de encontrar	41
Tabla 8. R6: Legible por todos los navegadores	41
Tabla 9. R7: Resolución mínima de 1024x768	42
Tabla 10. R8: Funcionalidad en los dispositivos móviles y tabletas.....	42
Tabla 11. R9: Banner	42
Tabla 12. R10: Breve descripción de la franquicia	42
Tabla 13. R11: Descripción grafica de los productos	43
Tabla 14. R12: Establecimientos.....	43
Tabla 15. R13: Listado de los establecimientos abiertos	43
Tabla 16. R14: Listado de los establecimientos que van abiertos próximamente	43
Tabla 17. R15: Noticias relacionadas.....	44
Tabla 18. R16: Listado de los establecimientos que van abiertos próximamente	44
Tabla 19. R17: Noticia destacada.....	44
Tabla 20. R18: Apartado de contacto.....	44
Tabla 21. R19: Envío de consultas.....	45
Tabla 22. R20: Apartado de la franquicia	45
Tabla 23. R21: Gestor	45
Tabla 24. R22: Acceso al gestor con nombre de usuario y contraseña	46
Tabla 25. R23: Gestión de los datos de los restaurantes	46
Tabla 26. R24: Gestión de las noticias.....	46
Tabla 27. R25: Gestión del banner	47
Tabla 28. R26: Gestión del banner	47
Tabla 29. R27: Hardware de usuario.....	47
Tabla 30. R28: Restricciones de usuario	47
Tabla 31. R29: Restricciones del servidor	48
Tabla 32. Usuario accede a la página web	53
Tabla 33. Dar de alta Gestores	54
Tabla 34. Borrar Gestores	54
Tabla 35. Tareas de mantenimiento	54
Tabla 36. Insertar en la tabla Banner.....	55
Tabla 37. Insertar en la tabla Restaurantes.....	55
Tabla 38. Insertar en la tabla Noticias.....	55
Tabla 39. Modificar en la tabla Banner.....	56
Tabla 40. Modificar en la tabla Noticias	56
Tabla 41. Modificar en la tabla Restaurantes	56
Tabla 42. Borrar en la tabla Banner	57
Tabla 43. Borrar en la tabla Restaurantes.....	57
Tabla 44. Borrar en la tabla Noticias.....	57
Tabla 45. Ítems obtenidos	71
Tabla 46. Valores de ítems.....	71
Tabla 47. Puntos de función sin ajustar.....	72
Tabla 48. Valores del factor de complejidad.....	73
Tabla 49. Factores de escala.....	75
Tabla 50. Valores factores de escala	75



Tabla 51 Valor de los Factores de costo	79
Tabla 52 Duración y coste estimados, por fases	82
Tabla 53 Pruebas visualización	187
Tabla 54 Pruebas login	187
Tabla 55 Pruebas en el gestor I	188
Tabla 56 Pruebas en el gestor II	188



1.- INTRODUCCIÓN

1.1.- Contexto

En este proyecto se va a desarrollar un portal web comercial, el cual estará formado principalmente por páginas web. Las páginas web son la herencia de lo que se denomina documentos de referencias cruzadas. Antes de la aparición de internet a nivel mundial ya existían los documentos con referencias cruzadas. El concepto en sí no es nuevo. Las referencias a otros documentos existían ya en los manuscritos medievales. La diferencia es que la Web es más global, más rápida, y más fácil de usar. Todo ello gracias a los avances tecnológicos de finales del siglo pasado.

En los años 60, Douglas Engelbart, mientras trabajaba en el Stanford Research Institute, propuso el NLS (oNLine System), un entorno de trabajo por computadora, con un sistema para almacenar publicaciones, con catálogos e índices para facilitar la búsqueda, y con reglas establecidas para citar documentos, de modo que fuera más fácil para los lectores acceder a los documentos referenciados. Era un entorno con teclado, pantalla, ratón e impresora, con posibilidad de teleconferencia y correo electrónico a través de una red de computadoras para una rápida comunicación entre los profesionales. Tenía las herramientas básicas de composición, estudio, organización y modificación de información. Los ficheros se guardaban jerárquicamente para su mejor organización. Se trabajaba con los documentos en modo multiventana, para ver varios documentos a la vez en ventanas diferentes, y se podían copiar objetos seleccionados de una ventana a otra.

El término "hipertexto" fue acuñado por Ted Nelson en 1965, en su artículo "A File Structure for the Complex, the Changing, and the Indeterminate", que leyó durante la vigésima conferencia anual de la Association of Computer Machinery (ACM). Ted Nelson ideó un modelo para la interconexión de documentos electrónicos. El proyecto Xanadu aún continúa luchando para conseguir un modelo de hipertexto superior al que trajo la World Wide Web.

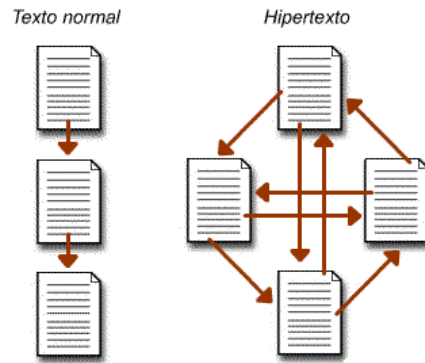


FIGURA 1 Hipertexto.

La World Wide Web fue inventada en 1989 por un informático del CERN (Organización Europea de Investigación Nuclear) llamado Tim Berners-Lee. Era un sistema de hipertexto para compartir información basado en Internet, concebido originalmente para servir como herramienta de comunicación entre los científicos nucleares del CERN. Tim Berners-Lee había estado experimentando con hipertexto desde 1980, año en que programó Enquire, un programa para almacenar piezas de información y enlazarlas entre ellas. Enquire se ejecutaba en un entorno multiusuario y permitía acceder a varias personas a los mismos datos. Tim Berners-Lee entregó su propuesta al CERN en 1989, en septiembre de 1990 recibió el visto bueno y junto con Robert Cailliau comenzó a escribir el nuevo sistema de hipertexto. A finales de 1990 el primer *browser* de la historia, WorldWide Web, ya tenía forma.

Los documentos necesitaban un formato que fuera adecuado para su misión. En aquella época casi todo el mundo utilizaba TeX y PostScript, pero éstos eran demasiado complicados teniendo en cuenta que debían ser leídos por todo tipo de computadoras, desde los terminales hasta las estaciones de trabajo gráficas X-Windows. Así, tanto el lenguaje de intercambio (HTML), como el protocolo de red (HTTP) se diseñaron para ser realmente muy simples.

HTML son las siglas de "*HyperText Mark-up Language*". "Mark-up" es un término de imprenta que significa *el conjunto de instrucciones estilísticas detalladas escritas en un manuscrito que debe ser tipografiado*. Así, HTML podría ser traducido como "*Lenguaje de Formato de Documentos para Hipertexto*". HTML es una aplicación de SGML, un lenguaje muy general para definir lenguajes de formato de documentos.

A principios de 1993 había alrededor de 50 servidores. Existían básicamente dos tipos de *browsers*: el original, gráfico, pero sólo para plataformas NeXT, y el browser en modo de línea, preparado para cualquier plataforma pero muy limitado y muy poco atractivo. En Febrero se lanzó la primera versión alfa del navegador "Mosaic for X", desarrollado en el NCSA (National Center for Supercomputing Applications).



FIGURA 2 NeXTcube usado por Berners-Leeen. El CERN Primer servidor web.

Funcionaba en X Windows, que era una plataforma popular entre la comunidad científica. En Abril el tráfico de la WWW era el 0,1% del total de Internet. El CERN declaraba la WWW como tecnología de acceso gratuito. En septiembre ya había versiones de Mosaic para PC y Macintosh. El tráfico alcanzaba el 1% de todo el tráfico de Internet y había más de 500 servidores. Es el comienzo del crecimiento explosivo de la Web. A finales del 94 ya había más de 10.000 servidores y 10 millones de usuarios. En 1997, más de 650.000 servidores.

Hoy la Web es algo cotidiano, sus utilidades son diversas (imágenes, vídeos, música, se pueden comprar cosas, se pueden hacer reservas...) y su impacto en la economía mundial es más que apreciable. La mayoría de empresas tienen su propia página web, haciéndose más competitivas. Hoy en día el marketing es imprescindible para cualquier empresa que quiera sobrevivir en un mundo tan competitivo como el nuestro.

El Marketing en Internet, objetivo de este proyecto, se originó a inicios de los 1990s como páginas web sencillas, de solo texto que ofrecían información de productos. Luego, siguiendo el desarrollo de internet, evolucionó en avisos publicitarios completos con gráficos.

Actualmente internet constituye un medio de comunicación cada día más importante y utilizado por más personas. Como sabemos, Internet es un medio interactivo, que a diferencia de los medios tradicionales como radio o televisión, permite conocer las preferencias y tendencias de consumo del posible cliente y desplegar información personalizada de acuerdo a ellas.

Internet ha cambiado mucho el mundo del marketing y de los negocios en general. Por eso, debemos de tener en cuenta el empleo de ese medio a la hora de elaborar nuestra estrategia de marketing. Actualmente, el e-Marketing en internet es una alternativa cada vez más utilizada por las empresas para difundir y promocionar sus productos y servicios. La verdadera importancia del marketing en Internet son las grandes posibilidades que ofrece, casi infinitas, siendo el presupuesto y la originalidad de los departamentos de marketing los que ponen el límite.

La publicidad en internet ofrece muchas ventajas:

- Permite medir de forma muy precisa y de forma continua el resultado de las campañas de publicidad.
- Permite entrar directamente en contacto con los potenciales clientes o usuarios.
- Permite continuas adaptaciones y modificaciones de las campañas según las exigencias del momento.
- Es mucho más económico que los medios de comunicación "tradicionales"
- Analizar el tipo de clientela que adquiere nuestro producto.

El eMarketing de buscadores y email son la base de las campañas de más éxito en la actualidad. No obstante, están tomando fuerza canales emergentes como los Blog's, las redes sociales y el RSS (Really Simple Syndication).

Es muy extraño que una sola estrategia te haga tener éxito en tu negocio en Internet. Hay que concretar una estrategia de marketing a varios niveles o incluso elegir la estrategia más adecuada. Hay muchas posibilidades: confección de una página web orientada a las ventas, marketing por email, optimización para buscadores (la mejor herramienta de publicidad y gratuita)....

1.2.- Motivación

En el ámbito empresarial, España ocupa el tercer puesto en el uso de Internet en Europa, con un 87% de empresas de más de 10 empleados conectadas a la Red, frente a una media del 75% de empresas. Solo un 35% de las microempresas españolas no están conectadas a Internet. Según ha puesto de manifiesto un estudio de Microsoft (http://www.microsoft.com/spain/empresas/tecnologia/empresas_conectadas.msp). En el caso de los autónomos, el porcentaje es sensiblemente inferior. Los motivos que se esconden tras este hecho suelen ser siempre los mismos: el presupuesto, el tiempo, o ambos.

A continuación se muestran los motivos por los cuales una empresa ha de tener su propia página web:

- **Expandir el negocio**

Tener una Página Web es la manera más barata de darse a conocer de manera continua sin hacer un gran esfuerzo de inversión en otro tipo de publicidad.

Con una Pagina Web la empresa estará presente en todo el mundo, podrá acceder a distintos mercados y conseguir nuevos clientes en lugares que quizás nunca haya imaginado.

- **Reconocimiento y Prestigio**

Hoy en día toda empresa de cualquier tamaño necesita tener una presencia en internet para dar una buena imagen de prosperidad y futuro.

- **Aumento de nuevos clientes**

Aumenta espectacularmente el conocimiento de su empresa o negocio, dándose a conocer por un inmenso número de empresas y personas (futuros clientes) no sólo locales, sino nacionales o incluso de todo el mundo.

Mediante las búsquedas que hacen los clientes en los buscadores de internet informándose de productos y servicios que necesitan, aparecen como resultados las páginas web de las empresas que ofrecen esos productos y servicios. Es decir, una empresa se da a conocer a otras personas y empresas que posiblemente se conviertan en nuevos clientes gracias a aparecer en un buscador de internet.

- **Fidelización de clientes existentes**

Puede dar una mejor imagen corporativa a los clientes que ya tiene mediante la comunicación de su nueva página web, y reducir el riesgo de pérdida de clientes por otros competidores que han encontrado a través de sus webs en internet.

Nace la posibilidad de comunicarse mejor con sus actuales clientes, informándoles constantemente de sus nuevos productos y servicios, sin las molestias de otro tipo de publicidad más agresiva.

- **Sus clientes pueden recomendarle más fácilmente:**

Si el producto es bueno, los clientes recomendarán dicho producto gustosamente con toda seguridad. En estos tiempos que corren, era del mundo online, se produce el fenómeno de la propagación a través de medios como el correo electrónico, Facebook u otras redes sociales, así como a través de los clásicos foros. Sin página Web es imposible que nadie enlace un sitio y la empresa sin página web renuncia a este importante, y aún en gran auge, canal de recomendación.

- **Mayor información y comunicación**

Se mejora la información y comunicación ofrecida a los clientes, a proveedores, a socios y colaboradores, empleados y entorno social.

Una página web ayuda mucho como un canal de comunicación adicional para informar sobre los productos y servicios que más interesa promocionar para aumentar la rentabilidad.

Los valores de la empresa o sus factores competitivos son conceptos que muchas veces no se comunican con eficacia por su complejidad. La comunicación mediante una web ayuda mucho a transmitir este tipo de valores e incluso muchas veces puede obligar a definirlos.

- **Aumento de ventas**

El aumento de nuevos clientes y la fidelización de los actuales clientes mediante mayor información y comunicación significan un incremento de las ventas y un mayor beneficio para el negocio.

- **Ahorro de costes en publicidad y comunicación**

Los costes de comunicación hacia el exterior se reducen al disponer de un sitio web donde se puede informar, comunicar, publicitar o promocionar una empresa, productos y servicios mediante un canal de comunicación abierto 24 horas al día durante los 365 días del año.

En cualquier momento se tiene la posibilidad de que alguien más le conozca vía internet. Esto es cada día más importante debido al aumento constante de todo tipo de personas y empresas que utilizan o empiezan ahora a utilizar internet.

- **Rápido retorno de la inversión**

Cualquiera de los puntos anteriores hace que el retorno de la inversión se consiga muy rápidamente al obtener rápidos beneficios con una baja inversión.

La inversión necesaria para ser parte de la red de internet mediante un sitio web es muy baja comparada con los beneficios que se obtienen en:

- comunicación e información
- marketing y publicidad
- mejora en la relación con los clientes
- ahorro de costes internos relacionados con el contacto con clientes
- imagen corporativa
- reconocimiento y prestigio
- expansión del negocio
- obtención y captación de nuevos clientes
- mantenimiento y fidelización de clientes existentes
- aumento de las ventas

1.3.- Definición del problema

Se quiere desarrollar un portal web con fines comerciales para franquicias. La idea es captar clientes, atraer su atención y a la vez informales de todos aquellos detalles que quieran saber sobre la franquicia para la que se quiere desarrollar el portal web comercial.

Para desarrollar la aplicación web se va a crear una franquicia ficticia. Esta franquicia nace de la idea de crear una franquicia de restaurantes destinados a la venta de sushi como complemento alternativo en la dieta europea. Pretende que el sushi forme parte de nuestra alimentación, ya que posee muchas cualidades beneficiosas para el organismo.

El problema que se nos plantea es la creación de la página web de la franquicia ficticia. La creación de la pagina comienza una vez se ha creado la imagen de la franquicia, de donde se deriva el diseño y los colores de la pagina web, para conseguir una línea de imagen corporativa en la misma y de esta manera conseguir que la página web mantenga una línea de imagen corporativa y un diseño que la otorgue cierta diferenciación al resto de las paginas web.

Se pretendía realizar una web con un estilo y diseño moderno, nuevo e impactante. Se pensó en que la navegación podría ser de izquierda a derecha y no de arriba hacia abajo, siendo esta última la navegación mas empleada para este tipo de paginas webs.

Además la web ha de ser dinámica, pudiendo realizar cambios en tiempo real, y no comunicándose con el servidor y redirigiendo al usuario a otra pagina. Para conseguirlo será necesario emplear diferentes lenguajes de programación web; Html, css, javascript, php, sql y jquery.

La pagina web se crea con fines publicitarios e informativos para los posibles clientes de la franquicia, tanto para los clientes de los restaurantes como para los posibles futuros propietarios de algún restaurante de la franquicia, es decir el publico objetivo de la web son los clientes de los restaurantes y los empresarios que deseen formar parte de la franquicia, o posibles franquiciados.



Ha de ser una página web atractiva para todos los usuarios que deseen saber más sobre esta franquicia. También es necesario que se informe de todos los aspectos de interés que ofrece dicha franquicia, inclinándose más por atraer a clientes a los restaurantes, por ser el público más numeroso.

Además la web debe tener cierto carácter asiático, estéticamente hablando, para no despistar al usuario.

1.4.- Estructura del documento

1.4.1.- Introducción

Esta memoria del proyecto de fin de carrera esta formada por varios apartados. A continuación se explicara en un breve resumen de que trata cada de ellos y los temas que los forman.

1.4.2.- Estructura del documento

Análisis

- Tecnologías web: Comparación, selección y uso de las diferentes tecnologías web para el desarrollo del portal web comercial.
- Ejemplos reales: Muestra de ejemplos de páginas web reales que utilizan algunos de los recursos o tienen alguna de las características empleadas en el desarrollo del portal web comercial.
- Requisitos: Condiciones necesarias para el desarrollo de la aplicación web. Se van a clasificar y especificar cada una de ellas.
- Casos de uso: descripción de los pasos o las actividades que deberán realizarse para llevar a cabo los procesos que ha de tener la aplicación web.
- Características de los usuarios: características de la interfaz para con los usuarios.
- Arquitectura de la base de datos: como y porque esta formada la base de datos.
- Planificación: estimación del coste y tiempos para el desarrollo del portal web comercial.
- Gráfico Grant: representación visual del tiempo y acciones necesarias para realizar el portal web, teniendo en cuenta las estimaciones obtenidas en la planificación.

Diseño

- Disposición de las páginas: se muestran y comentan los patrones de diseño que se han seguido.
- Diseño de la base de datos: como esta estructurada la base de datos y porque.

Implementación

- La implementación se ha realizado en 3 capas (capa de presentación, capa de negocio y capa de datos), se muestra cada una de ellas, porque esta formada, cual es su cometido, como funciona y porque.

Batería de pruebas

- Se muestra un resumen de las pruebas realizadas, los resultados obtenidos y se explica porque se han realizado dichas pruebas

Conclusiones

- Conclusiones obtenidas después de haber desarrollado el portal web comercial de este proyecto de fin de carrera.

Anexos

- Presupuesto final del proyecto.
- Explicación para la instalación del portal web comercial.
- Manual de usuario.

Bibliografía

- Paginas web y libros empleados para el desarrollo del portal web comercial y la memoria.



2.- ANÁLISIS

2.1.- Tecnologías web

2.1.1.- Introducción

En este apartado se van a ver los tipos de lenguajes mas empleados en el desarrollo de aplicaciones web, sus características, cuales se han elegido, porque se han elegido y cual es su cometido en la aplicación web.

También serán vistos algunos ejemplos de páginas web dinámicas, comentando los aspectos que mas interesan para el desarrollo del portal web comercial tratado en este proyecto de fin de carrera.

2.1.2.- Comparativa de Tecnologías web

Existen múltiples lenguajes de programación para el desarrollo de aplicaciones web, cada una de ellas con sus características propias, sus ventajas y sus desventajas. A continuación se muestra una tabla en la que se comparan algunas de las propiedades de los lenguajes de programación web más empleados en el desarrollo de aplicaciones web:

Lenguajes de programación web	Software libre	Soporte para bases de datos	Soporte para paginas web dinámicas	Manipulación de formularios
PHP	✓	✓	✓	✓
ASP.NET	✗	✓	✓	✓
JavaScript	✓	✓	✓	✓
jQuery	✓	✓	✓	✓
Ajax	✓	✓	✓	✓
JSP	✗	✓	✓	✓
Flash	✗	✓	✓	✓

Lenguajes de programación web	Manipulación de cookies	Múltiples librerías	Funcionamiento en todos navegadores
PHP	✓	✓	✓
ASP.NET	✓	✓	✗
JavaScript	✓	✓	✓
jQuery	✓	✓	✓
Ajax	✓	✓	✓
JSP	✓	✓	✓
Flash	Emplea las cookie flash, o Local Shared Object (LSO), es una colección de archivos tipo cookie almacenados como archivo en computador del usuario, pero no son las cookies que emplean los navegadores	✓	✗

Tabla 1. Comparación de tecnologías web

PHP es uno de los lenguajes mas completos, igual que ASP.NET, la diferencia entre ambos es que PHP es software libre y por lo tanto no es necesario pagar por el para desarrollar un proyecto. Además ASP.Net tiene el enorme problema de que solo corre en plataformas con el sistema operativo Windows. Por lo tanto se empleara PHP.

JSP tampoco es software libre, por lo tanto se descarta

Se va a emplear jQuery para la creación de acciones dinámicas. En este ámbito se descarta Ajax porque jQuery ya integra funcionalidades para trabajar con AJAX. Para usar jQuery es necesario el uso de JavaScript, ya que el propio jQuery emplea funcionalidades JavaScript.

Se va usar Flash para algún detalle estético de la web, no para realizar ninguna función en concreto por ser un lenguaje no reconocido en iPhone, ipod e ipad.

Por lo tanto se empleara los lenguajes PHP, JavaScript, jQuery y Flash.

Para la generación del cuerpo de las paginas se va a emplear HTML, y para dar estilo a los elementos de las páginas web se va a emplear CSS.

Existen plataformas que generan el código de páginas web, tales como Wordpress y Drupal. El problema de estas plataformas es la escasa flexibilidad que presentan si lo comparamos con la creación desde 0 de páginas web mediante código web puro. Estas plataformas pretenden ahorrar esfuerzos al programador generando el código y demás elementos que se necesitan para el desarrollo de la página web, lo que limita las posibilidades de creación, y en ocasiones generan código incensario.

En cuanto a la base de datos existen múltiples posibilidades. Durante la carrera se han visto bases de datos sobre SQL, por lo tanto se van a comparar las plataformas SQL mas utilizadas en el desarrollo de plataformas web.

Característica	MySQL	SQL Server Express	SQL Server
Costo	Libre y de pago	Libre	De pago
Open Source	Si	No	No
Plataformas	Linux, Windows y muchas otras	Sólo Windows	Sólo Windows
Limite de tamaño de la base de datos	Limitado por el sistema operativo	10Gb	Limitado por el sistema operativo
Compatibilidad ACID	Depende del motor de almacenamiento	Si	Si
Transacciones	Si	Si	Si
Servicio de reportes	No	Si	Si
Posibilidad de elegir diferentes formas de almacenamiento	Si	No	No
Vistas	Si	Si	Si
Procedimientos almacenados	Si	Si	Si
Triggers	Si	Si	Si
Cursores	Si	Si	Si
Subconsultas	Si	Si	Si
Replicación	Si	Limitado	Si
Funciones definidas por el usuario (UDF)	Si	Si	Si

Tabla 2. Comparación de bases de datos SQL

Se ha decidido usar MySQL por sus ventajas frente a sus homólogos. La principal característica, la cual no aparece en la tabla, es que PHP está asociado con MySQL. Además es Open Source, reduciendo los gastos a 0, trabaja sobre la gran mayoría de Sistemas Operativos y el tamaño de la base de datos viene determinado por el sistema operativo, en el caso de Linux se puede determinar el tamaño de la base de datos.

2.1.3.- Tecnologías web seleccionadas

Html

La primera descripción de HTML disponible públicamente fue un documento llamado HTML Tags (Etiquetas HTML), publicado por primera vez en Internet por Tim Berners-Lee en 1991. Describe 22 elementos comprendiendo el diseño inicial y relativamente simple de HTML. Trece de estos elementos todavía existen en HTML 4.

Berners-Lee consideraba a HTML una ampliación de SGML, pero no fue formalmente reconocida como tal hasta la publicación de mediados de 1993, por la IETF, de una primera proposición para una especificación de HTML: el boceto Hypertext Markup Language de Berners-Lee y Dan Connolly, el cual incluía una Definición de Tipo de Documento SGML para definir la gramática. El boceto expiró pasados seis meses, pero fue notable por su reconocimiento de la etiqueta propia del navegador Mosaic usada para insertar imágenes sin cambio de línea, reflejando la filosofía del IETF de basar estándares en prototipos con éxito. Similarmente, el boceto competidor de Dave Raggett HTML+ (Hypertext Markup Format) (Formato de marcaje de hipertexto), de 1993 tardío, sugería, estandarizar características ya implementadas tales como tablas.

HTML es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Todos los navegadores de hoy en día interpretan código HTML. Además es un lenguaje sencillo y que apenas ocupa espacio, haciendo que los tiempos de carga sean rápidos.

El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). En HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Css

CSS se ha creado en varios niveles y perfiles. Cada nivel de CSS se construye sobre el anterior, generalmente añadiendo funciones al previo. Los perfiles son, generalmente, parte de uno o varios niveles de CSS definidos para un dispositivo o interfaz particular. Actualmente, pueden usarse perfiles para dispositivos móviles, impresoras o televisiones.

La primera especificación oficial de CSS, recomendada por la W3C fue CSS1, publicada en diciembre 1996, y abandonada en abril de 2008, seguida por varias versiones. Actualmente la última versión de CSS es CSS 3.

Con Css se evita escribir código incensario en html referente al estilo. Al centralizar todo el estilo de la pagina web en una o varias hojas de estilo Css facilita la gestión del estilo de toda la web y sus componentes, siendo esto muy útil para cambiar un aspecto del diseño de un sitio web, sin hojas de estilo se tendría que cambiar página a página dicho aspecto.

Nos permite definir aspectos concretos de un documento, facilitando el diseño de los documentos. Html se queda cortó en algunos aspectos de diseño. Se pueden definir los estilos de tal manera que será diferente la visualización en una PDA que en un PC, o también hojas de estilo especiales para impresión (sin banners ni imágenes de publicidad por ejemplo) o para ser leída por un sintetizador de voz.

Se pueden colocar elementos con mayor precisión, sin hojas de estilo se suele recurrir a tablas para la maquetación, pese a que esto implica un problema de accesibilidad (los contenidos para una persona ciega en realidad están desordenados). Además es complicado lograr el punto exacto donde se quiere colocar algo y se genera una cantidad de código excesiva.

El sitio web será más respetuoso con los estándares de desarrollo web con el uso de CSS, lo que implica que será más funcional en los diferentes navegadores. Tiene la ventaja de ser un lenguaje reconocido por todos los navegadores.

Se logra reducir el peso de los ficheros, disminuyendo el ancho de banda que consumes. Además ayuda a los robots de los motores de búsqueda. Definiendo el diseño en la hoja de estilo se ahorra trabajo a los buscadores, ya que podrá indexar documentos con mayor rapidez. Se logra aumentar la densidad de las palabras clave ya que se reduce la utilización de etiquetas.

JavaScript



FIGURA 3. Logo JavaScript

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.003 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

«JavaScript» es una marca registrada de Oracle Corporation. Es usada con licencia por los productos creados por Netscape Communications y entidades actuales como la Fundación Mozilla.

Microsoft dio como nombre a su dialecto de JavaScript «JScript», para evitar problemas relacionados con la marca. JScript fue adoptado en la versión 3.0 de Internet Explorer, liberado en agosto de 1996. Los dialectos pueden parecer tan similares que los términos «JavaScript» y «JScript» a menudo se utilizan indistintamente, pero la especificación de JScript es incompatible con la de ECMA (European Computer Manufacturers 'Association) en muchos aspectos.

Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar Document Object Model (DOM, o Modelo de Objetos del Documento en español).

En 1997 los autores propusieron JavaScript para que fuera adoptado como estándar de la European Computer Manufacturers 'Association ECMA, que a pesar de su nombre no es europeo sino internacional, con sede en Ginebra. En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también como un estándar ISO.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador... aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Php



FIGURA 4. Logo PHP

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools.

El PHP es, probablemente, el lenguaje que más evoluciona actualmente. Es de libre distribución y desde su versión 4, pese a ser un lenguaje interpretado y bastante rápido.

Es un lenguaje Rápido (pese a ser interpretado), multiplataforma, multitud de librerías reutilizables. Se puede instalar un servidor gratuito (Apache).

Es un lenguaje asociado a MySQL (BD de libre distribución) proporciona gran potencia en creación de páginas web, de forma rápida, fácil y barata (gratis).

Las razones de utilizar PHP se deben a su poder y sencillez. Una de las principales razones es que PHP es un software libre, y una de sus grandes cualidades es su versatilidad al momento de escribir código, su sencillez en la sintaxis e inclusive su seguridad.

Posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de nuestros programas.

PHP es un lenguaje de uso muy común en la web, además de ser libre, esto significa que una inmensa comunidad de programadores que utilizan este lenguaje están cooperando para la mejora del motor de PHP, por lo cual es cada vez más seguro y estable a medida que pasa el tiempo y aumenta su versión.

Una de las razones es el soporte para bases de datos, envío de e-mails y dinamismo en páginas web, que a pesar de que también encontramos estas características en otros lenguajes, php es fácil de implementar, seguro y gratuito.

MySql



FIGURA 5. Logo MySQL

SQL («lenguaje de consulta estructurado») fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde entonces ha sido considerado como un estándar para las bases de datos relacionales.

Desde 1986, el estándar SQL ha aparecido en diferentes versiones. MySQL es una idea originaria de la empresa opensource MySQL AB establecida inicialmente en Suecia en 1995 y cuyos fundadores son David Axmark, Allan Larsson, y Michael "Monty" Widenius. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

Michael «Monty» Widenius en la década de 1990 trató de usar mSQL para conectar las tablas usando rutinas de bajo nivel ISAM; sin embargo, mSQL no era rápido y ni flexible para sus necesidades. Esto lo llevó a crear una interfaz de programación de aplicaciones SQL denominada MySQL para bases de datos muy similar a la de mSQL pero más portable.

El nombre de MySQL procede de la combinación de My, hija Widenius, con el acrónimo SQL (según la documentación de la última versión en inglés).

MySql también es código abierto, MySQL usa la licencia GPL, y por lo tanto gratuito. Es un lenguaje que ofrece velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento. Tiene bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema. Es facilidad de configurar e instalar, soporta gran variedad de Sistemas Operativos.

Tiene la característica de tener una baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

Por lo tanto su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder a bases de datos en Internet.

jQuery



FIGURA 6. Logo jQuery

Es un lenguaje sencillo, si lo comparamos con sus homólogos y dispone de un amplio conjunto de funciones para animar el contenido de la página de forma muy sencilla.

Su lema "escribe menos, haz más" describe perfectamente una de las principales características de este lenguaje. La intención es ahorrar código a los programadores creando funciones sencillas que realizan varias tareas. Se pueden anidar algunas funciones, es decir en la misma línea de código podemos manipular el estilo de una etiqueta y animarla, por ejemplo hacer visible una imagen y animarla.

Es compatible por los navegadores actuales.

Por defecto integra funcionalidades para trabajar con AJAX, lo cual hace posible que la página sea más dinámica.

Tiene la ventaja de que el servidor lo enviará al cliente la primera vez que visite una página del sitio. En siguientes páginas el cliente ya tendrá el archivo de jQuery, por lo que no necesitará transferirlo y lo tomará de la caché. Con lo que la carga de la página sólo se verá afectada por el peso de este framework de jQuery una vez por usuario.

jQuery es un producto con gran aceptación por parte de los programadores y tiene un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones.

2.1.4.- Empleo de tecnologías web

Las tecnologías que se van a emplear en la creación de esta página web son un servidor apache alojado en un servidor o hosting. Dicho servidor contendrá además la base de datos MySQL, la cual puede ser administrada manualmente a través de PHPMyAdmin, o automáticamente a través de un gestor de contenido de la propia pagina web, el cual se comunicara con el servidor mediante sentencias PHP.

En cuanto al contenido de la programación se usaran varios lenguajes de programación; Html, Css, JavaScript, jQuery, PHP y sentencias SQL para crear y manejar la base de datos. A continuación voy a explicar brevemente cual será la función de cada uno de los lenguajes de programación comentados en el apartado anterior.

Html será empleado para crear la estructura de la página web. Incluirá la parte del contenido que es fija, es decir, que siempre ha de estar presente en la web. No solo incluirá contenido de texto, también contendrá el formulario para las sugerencias de los usuarios, imágenes, titulares... Es el lenguaje, junto con css, que se utilizara para realizar toda la maquetación de la página web, siguiendo estrictamente el diseño que previamente habrá sido aprobado por el cliente

CSS es el lenguaje a emplear para crear y modificar el estilo de la web; tipos de letras, colores, bordes, posición de lo elementos... Es uno de los lenguajes más empleados en materia web para manejar el estilo de las mismas.

JavaScript, este lenguaje efectuara mucha de las funcionalidades de la web, será el encargado de llamar a las funciones de jQuery, dar funcionalidad a muchos de los botones de la web, gestionara los banners o rotadores que se encuentran en el comienzo de la página web...

Php será el lenguaje encargado de, entre otras cosas, comunicarse con la base de datos, cargando el contenido necesario, realizando las consultas necesarias, eliminando contenido cuando es necesario, modificarlo... También se encargara de maquetar y enviar el mail de consulta de un usuario a su destinatario. Además tendrá una función muy importante, reconocer el dispositivo que lea la pagina web, en el caso de que dicho dispositivo sea un Smartphone, tableta electrónica o iPod modificara los estilos de la pagina para que no solo sea visible por todos los dispositivos si no que además optimizara la pagina web para ese dispositivo en concreto y así conseguir el aspecto visual deseado.

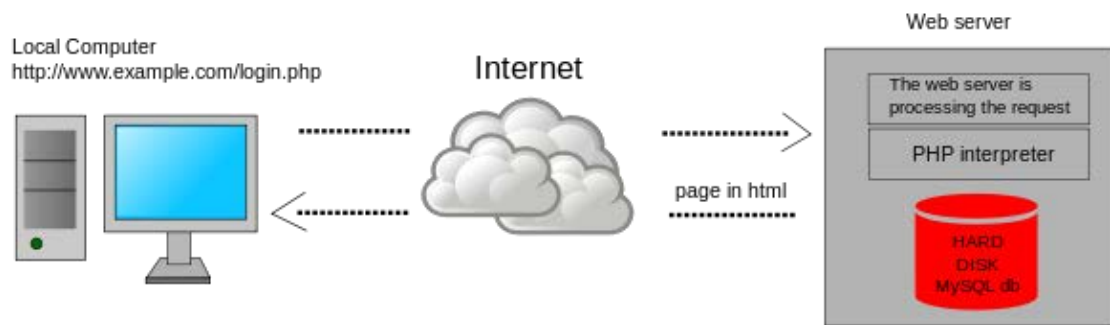


FIGURA 7. Estructura del portal web comercial

MySQL se utilizara para crear la base de datos que alberga toda la información susceptible de cambios. También se empleara para modificar contenido, borrarlo y realizar consultas a la base de datos.

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Es el lenguaje que aportara a la web la propiedad de ser dinámica, pudiendo realizar cambios en la misma sin cambiar de página.

Flash CS4 será usado para crear pequeñas animaciones a modo de decoración en la web que separaran una sección de otra.

Se ha tenido en cuenta que algunos Smartphone no disponen de Flash, la solución la dará PHP mediante sentencias que identifican el explorador y el sistema operativo que lo aloja y eliminando estas animaciones de la página web en caso de que sea un dispositivo sin Flash.

2.2.- Ejemplos reales

2.2.1.- Introducción

Se van a mostrar algunos ejemplos de páginas web con navegación horizontal, demostrando que es un tipo de navegación utilizada en portales web, y algunos ejemplos de páginas web dinámicas, en las cuales el contenido no es fijo sino que se construye de acuerdo a la interacción que el usuario hace con la página, en caso de este portal web comercial el usuario gestor es que interactúa con el gestor, según como interactúe con el gestor modificara parte del contenido de la página web principal.

2.2.2.- Ejemplos

Kia Picanto

Página web con navegación horizontal y con un diseño destacable. La navegación es muy sencilla, cada sección consta de una o dos flechas para pasar a la siguiente sección o a la anterior. Además posee un menú en la parte inferior para ir a cada sección independientemente. Es un sistema muy intuitivo y fácil de usar gracias a la buena administración del contenido.

El desplazamiento lateral se realiza mediante el empleo de la librería `jquery.scrollTo.js`.

<http://www.kiapicanto.co.za/>



FIGURA 8 Página web de Kia Picanto

Gudanghome

Esta página también cuenta con navegación horizontal mediante el uso de jQuery para ir de una sección a otra desplazando el contenido hasta llegar al punto indicando. En esta ocasión la barra del navegador que aparece en la parte inferior del mismo no esta, ha sido eliminada mediante programación obligando al usuario a utilizar el menú de la parte inferior de la web. Esta solución es buena pero tiene un inconveniente, en caso de que el usuario no se percate del menú no podrá navegar por la web y muy posiblemente cerrara la página.

<http://www.gudanghome.com>



FIGURA 9 Página web de gudanghome

Amazon

Página web dinámica de venta de artículos por internet. Es una de las páginas web dinámicas mas conocidas por los usuarios de la red. La página web que se va a desarrollar en este proyecto de fin de carrera también es una página web dinámica.

Definimos página web dinámica como aquella pagina web donde el contenido no es fijo sino que se construye de acuerdo a la interacción que el usuario hace con la página. La información de este tipo de página suele estar almacenada en Bases de Datos de las cuales se extrae una parte según las selecciones o acciones llevadas a cabo por la persona que visita la página web.

<http://www.amazon.com/>

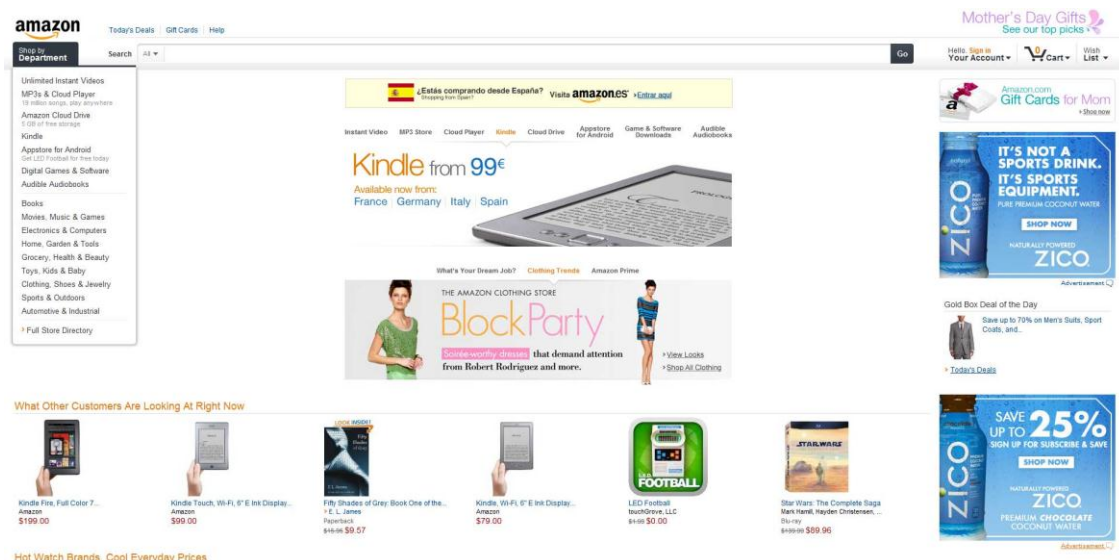


FIGURA 10 Página web de Amazon

2.3.- Requisitos

2.3.1.- Introducción

Se van a clasificar los requisitos por tipo y subtipo. Los tipos y subtipos disponibles son los siguientes:

- **Funcionalidad:** el requisito especifica una funcionalidad del sistema. Este requisito tendrá una estrecha relación con la interacción del usuario con el sistema. Los subtipos de los que dispone son los siguientes:
 - **Básica:** este subtipo indica que el requisito especificará funcionalidad básica, es un requisito que hay que cumplir estrictamente en el desarrollo de la aplicación.
 - **Adicional:** indica que el requisito especificará funcionalidad adicional, es un requisito que debe cumplirse en el desarrollo de la aplicación, pero tiene menos importancia que los requisitos funcionales básicos.
 - **Gráficas y documentos:** indica que el requisito especificará funcionalidad correspondiente a la realización de gráficas y generación de documentos.
 - **Extra:** indica que el requisito especificará funcionalidad extra.
- **Interfaces:** un requisito de este tipo especifica las conexiones de entrada/salida de los datos al/del sistema. Existen varios subtipos para este tipo como “De sistema”, “Externas”, “De usuario”, “Comunicaciones”, “Hardware” o “Software.
- **Restricciones de diseño:** los requisitos de este tipo especifican limitaciones de diseño que el sistema debe cumplir en el momento de su construcción.
- **Seguridad:** los requisitos de seguridad indican controles o medidas de seguridad que debe implantar o respetar el sistema para asegurar la integridad, disponibilidad y confidencialidad de los datos tratados y almacenados.

De forma adicional, por cada requisito se indicará el nivel de impacto y el nivel de esfuerzo. El impacto indica la importancia que tiene el requisito a la hora de su implantación al sistema actual. El esfuerzo indica la complejidad para cumplir dicho objetivo. Los niveles de medida serán: “Muy bajo”, “Bajo”, “Medio”, “Alto” y “Muy alto”.

Por otra parte, se indicará la versión final de cada requisito. Cada vez que el requisito ha sido editado, su versión ha aumentado en una unidad.

2.3.2.- Requisitos del sistema:

R1: Web impactante		
Descripción	Uno de los primeros requisitos que se planteo fue una web con un estilo y diseño moderno, nuevo e impactante. Se pensó en que la navegación podría ser de izquierda a derecha y no de arriba hacia bajo, siendo esta última la navegación estándar de las webs. Más tarde nos centraremos en este punto.	
Tipo:	Restricciones de diseño	Subtipo: -
Impacto:	Medio	Esfuerzo: Alto
Versión:	3	

Tabla 3. R1: Web impactante

R2: Fácil de navegar		
Descripción	La web ha de ser navegable por los usuarios, de lo contrario la página no será visitada y será inútil.	
Tipo:	Restricciones de diseño	Subtipo: -
Impacto:	Muy alto	Esfuerzo: Medio
Versión:	1	

Tabla 4. R2: Fácil de navegar

R3: Tipografía de la letra legible y con el estilo de la web		
Descripción	La tipografía ha de tener un estilo que el usuario identifique con el servicio que se ofrece, en este caso comida oriental. Tiene que ser legible para facilitar una lectura rápida y cómoda.	
Tipo:	Restricciones de diseño	Subtipo: -
Impacto:	Bajo	Esfuerzo: Bajo
Versión:	1	

Tabla 5. R3: Tipografía de la letra legible y con el estilo de la web

R4: Diseño atractivo	
Descripción	El diseño de la pagina web a de ser un diseño atractivo, que sea capaz de atraer a los usuarios. Es otra de las razones por las que se emplea la navegación lateral.
Tipo:	Restricciones de diseño Subtipo: -
Impacto:	Alto Esfuerzo: Alto
Versión:	4

Tabla 6. R4: Diseño atractivo

R5: Contenido fácil de encontrar	
Descripción	Cualquier contenido de la web tiene que ser fácil de localizar por los usuarios, debe ser accesible desde cualquier parte de la web, evitando pasar por varias páginas para llegar al contenido.
Tipo:	Restricciones de diseño Subtipo: -
Impacto:	Alto Esfuerzo: Alto
Versión:	3

Tabla 7. R5: Contenido fácil de encontrar

R6: Legible por todos los navegadores	
Descripción	Debe funcionar a la perfección en todos los navegadores actuales, exceptuando internet Explorer 6 y anteriores debido a su falta de recursos, lo que deriva en más esfuerzo, más tiempo y en un incremento del coste del proyecto. Hay que tener en cuenta que las versión 6 de internet Explorer y anteriores no reconocen sentencias css (por ejemplo: min-width) ni tampoco saben como manejar las transparencias de las imágenes con extensión .png.
Tipo:	Funcionalidad Subtipo: Básica
Impacto:	Alto Esfuerzo: Medio
Versión:	2

Tabla 8. R6: Legible por todos los navegadores

R7: Resolución mínima de 1024x768	
Descripción	La pagina web tiene que poder verse en una resolución de como mínimo 1024pixles x 768 pixeles, independientemente del navegador usado.
Tipo:	Restricciones de diseño Subtipo: -
Impacto:	Alto Esfuerzo: Medio
Versión:	2

Tabla 9. R7: Resolución mínima de 1024x768

R8: Funcionalidad en los dispositivos móviles y tabletas	
Descripción	La página web ha de funcionar en todos los smartphones y tabletas del mercado
Tipo:	Funcionalidad Subtipo: Básica
Impacto:	Bajo Esfuerzo: Medio
Versión:	1

Tabla 10. R8: Funcionalidad en los dispositivos móviles y tabletas

R9: Banner	
Descripción	La pagina web a de contener un banner en donde se muestran diapositivas. Cada cierto tiempo el banner a de cambiar de dispositiva.
Tipo:	Funcionalidad Subtipo: Básica
Impacto:	Medio Esfuerzo: Medio
Versión:	1

Tabla 11. R9: Banner

R10: Breve descripción de la franquicia	
Descripción	La pagina web debe tener un apartado donde se describen los productos que ofrece la franquicia, sus beneficios y ventajas sobre otras franquicias.
Tipo:	Funcionalidad Subtipo: Básica
Impacto:	Medio Esfuerzo: Bajo
Versión:	1

Tabla 12. R10: Breve descripción de la franquicia

R11: Descripción grafica de los productos			
Descripción	Que productos ofrece la franquicia, sus beneficios y ventajas sobre otras franquicias, explicado mediante un grafico.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Bajo
Versión:	1		

Tabla 13. R11: Descripción grafica de los productos

R12: Establecimientos			
Descripción	La página web a de tener una sección para los establecimientos de la franquicia, en donde se informe de la situación en donde se encuentran y los horarios tanto de apertura y como de cierre.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión:	2		

Tabla 14. R12: Establecimientos

R13: Listado de los establecimientos abiertos			
Descripción	Listado de los establecimientos abiertos de la franquicia.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión:	2		

Tabla 15. R13: Listado de los establecimientos abiertos

R14: Listado de los establecimientos que van a ser abiertos próximamente			
Descripción	Listado de los lugares donde se van a abrir nuevos establecimientos de la franquicia.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Bajo	Esfuerzo:	Medio
Versión:	3		

Tabla 16. R14: Listado de los establecimientos que van abiertos próximamente

R15: Noticias relacionadas			
Descripción	Conjunto de noticias que mantengan relación con la franquicia o los productos que ofrece.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión:	2		

Tabla 17. R15: Noticias relacionadas

R16: Listado de las noticias			
Descripción	Listado con los titulares de las noticias, ordenadas cronológicamente, en donde el usuario puede seleccionar la noticia que desea ver.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión:	1		

Tabla 18. R16: Listado de los establecimientos que van abiertos próximamente

R17: Noticia destacada			
Descripción	Noticia que se tiene que mostrar al cargar la página web.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Bajo
Versión:	1		

Tabla 19. R17: Noticia destacada

R18: Apartado de contacto			
Descripción	En esta sección se ha de mostrar los datos de contacto y ha de contener un formulario para mandar dudas o comentarios.		
Tipo:	Funcionalidad	Subtipo:	Básico
Impacto:	Medio	Esfuerzo:	Bajo
Versión:	1		

Tabla 20. R18: Apartado de contacto

R19: Envío de consultas	
Descripción	En la sección de contacto el usuario podrá rellenar y enviar el formulario de consultas, el cual se tiene que enviar al correo de la franquicia.
Tipo:	Interfaces Subtipo: Del sistema
Impacto:	Medio Esfuerzo: Medio
Versión:	3

Tabla 21. R19: Envío de consultas

R20: Apartado de la franquicia	
Descripción	En donde los usuarios que estén interesados en formar parte de la franquicia podrán informarse.
Tipo:	Funcionalidad Subtipo: Básico
Impacto:	Medio Esfuerzo: Bajo
Versión:	1

Tabla 22. R20: Apartado de la franquicia

R21: Gestor	
Descripción	Sección del portal web comercial donde los usuarios dados de alta como usuarios gestores podrán crear, eliminar o modificar contenido de parte de las secciones de la página web.
Tipo:	Funcionalidad Subtipo: Básico
Impacto:	Muy alto Esfuerzo: Muy alto
Versión:	3

Tabla 23. R21: Gestor

R22: Acceso al gestor con nombre de usuario y contraseña		
Descripción	<p>Para acceder al gestor será necesario logarse con una cuenta de administrador, a través de un email (empleado como nombre de usuario) y una contraseña.</p> <p>Será muy importante que el sistema proteja los datos de las diferentes secciones del sistema, no pudiendo acceder a éstos, salvo aquellas personas con autorización explícita por parte del administrador del sistema.</p>	
Tipo:	Seguridad	Subtipo: -
Impacto:	Alto	Esfuerzo: Medio
Versión:	1	

Tabla 24. R22: Acceso al gestor con nombre de usuario y contraseña

R23: Gestión de los datos de los restaurantes		
Descripción	<p>El gestor de restaurantes ha de ser capaz de crear nuevos restaurantes, tanto restaurantes ya abiertos como futuros restaurantes. En función del tipo de restaurante que sea, abierto o no, se han de rellenar todos los campos o solo algunos.</p> <p>También se debe poder modificar los datos de los restaurantes ya existentes en el sistema, asique como poder borrarlos.</p>	
Tipo:	Funcionalidad	Subtipo: Básico
Impacto:	Alto	Esfuerzo: Alto
Versión:	2	

Tabla 25. R23: Gestión de los datos de los restaurantes

R24: Gestión de las noticias		
Descripción	<p>Se ha de poder crear nuevas noticias, modificar las existentes en el sistema, borrar noticias en el sistema y debe existir la opción de poder seleccionar cual va a ser la noticia destacada, es decir la noticia que se muestra por defecto.</p>	
Tipo:	Funcionalidad	Subtipo: Básico
Impacto:	Alto	Esfuerzo: Alto
Versión:	1	

Tabla 26. R24: Gestión de las noticias

R25: Gestión del banner	
Descripción	Se ha de poder crear nuevas diapositivas, modificar las existentes en el sistema y borrar diapositivas que se encuentren en el sistema.
Tipo:	Funcionalidad Subtipo: Básico
Impacto:	Alto Esfuerzo: Alto
Versión:	1

Tabla 27. R25: Gestión del banner

R26: Seguridad frente ataques	
Descripción	El sistema deberá proteger los datos sensibles a ataques como Cross scripting o SQL Injection.
Tipo:	Seguridad Subtipo: -
Impacto:	Medio Esfuerzo: Alto
Versión:	1

Tabla 28. R26: Gestión del banner

R27: Hardware de usuario	
Descripción	El usuario deberá disponer de conexión a internet para acceder al sistema.
Tipo:	Restricciones de diseño Subtipo: -
Impacto:	Bajo Esfuerzo: Bajo
Versión:	1

Tabla 29. R27: Hardware de usuario

R28: Restricciones de usuario	
Descripción	El usuario deberá disponer de un navegador de internet que soporte HTML4, Javascript y CSS3.
Tipo:	Restricciones de diseño Subtipo: -
Impacto:	Bajo Esfuerzo: Bajo
Versión:	1

Tabla 30. R28: Restricciones de usuario

R29: Restricciones del servidor		
Descripción	<p>El servidor que ofrezca la aplicación deberá cumplir con unos requerimientos mínimos:</p> <ul style="list-style-type: none">• Tener instalado un sistema operativo Linux o Windows, cuya distribución esté orientada a servir aplicaciones.• Servidor web Apache con módulo PHP 5.3.9 (como mínimo) instalado.• Servicio MySQL 5.5.20 (como mínimo) instalado.• Almacenamiento recomendado: 2TB.• Ancho de banda recomendado: 100Mbits por segundo.• Memoria RAM recomendada: 16Gb.• Disponer de acceso FTP para la carga de los ficheros de la aplicación.	
Tipo:	Restricciones de diseño	Subtipo: -
Impacto:	Bajo	Esfuerzo: Medio
Versión:	2	

Tabla 31. R29: Restricciones del servidor

2.4.- Casos de uso

2.4.1.- Introducción

El diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema frente a la interacción externa.

2.4.2.- Tipos de usuario que interactúan con la aplicación

En la página web hay tres tipos de usuarios que pueden interactuar con la aplicación web.

- Usuario: consulta información en la base de datos.
- Gestor: realiza consultas, inserta, modifica y borrar registros en las tablas correspondientes a las diferentes secciones de la pagina web.
- Administrador: da de alta y borra a los gestores, realiza tareas de mantenimiento en la base de datos.

El usuario accede a la base de datos de manera transparente para el. Cuando un usuario accede a la página web esta realiza varias consultas a la base de datos para mostrar gran parte del contenido. Por lo tanto aunque el usuario sea transparente a la consulta es la razón por la que se realizan dichas consultas, por eso se considera un tipo de usuario que accede a la aplicación.

El Gestor es la persona que puede acceder a la base de datos a través de gestor, accede al gestor a través de un usuario y contraseña, puede realizar cualquier gestión sobre el contenido de la página web. Realiza consultas, inserta, modifica y borrar datos de las tablas que corresponden a las diferentes secciones de la pagina web. La única tabla a la que no puede acceder es la tabla de usuarios.

El usuario administrador es el único que puede acceder a todo el contenido de la base de datos. Su función principal es dar de alta y borrar a los usuarios gestores. También puede realizar tareas de mantenimiento, tales como reparar errores, realizar backups de la base datos, etc.

2.4.3.- Casos de uso

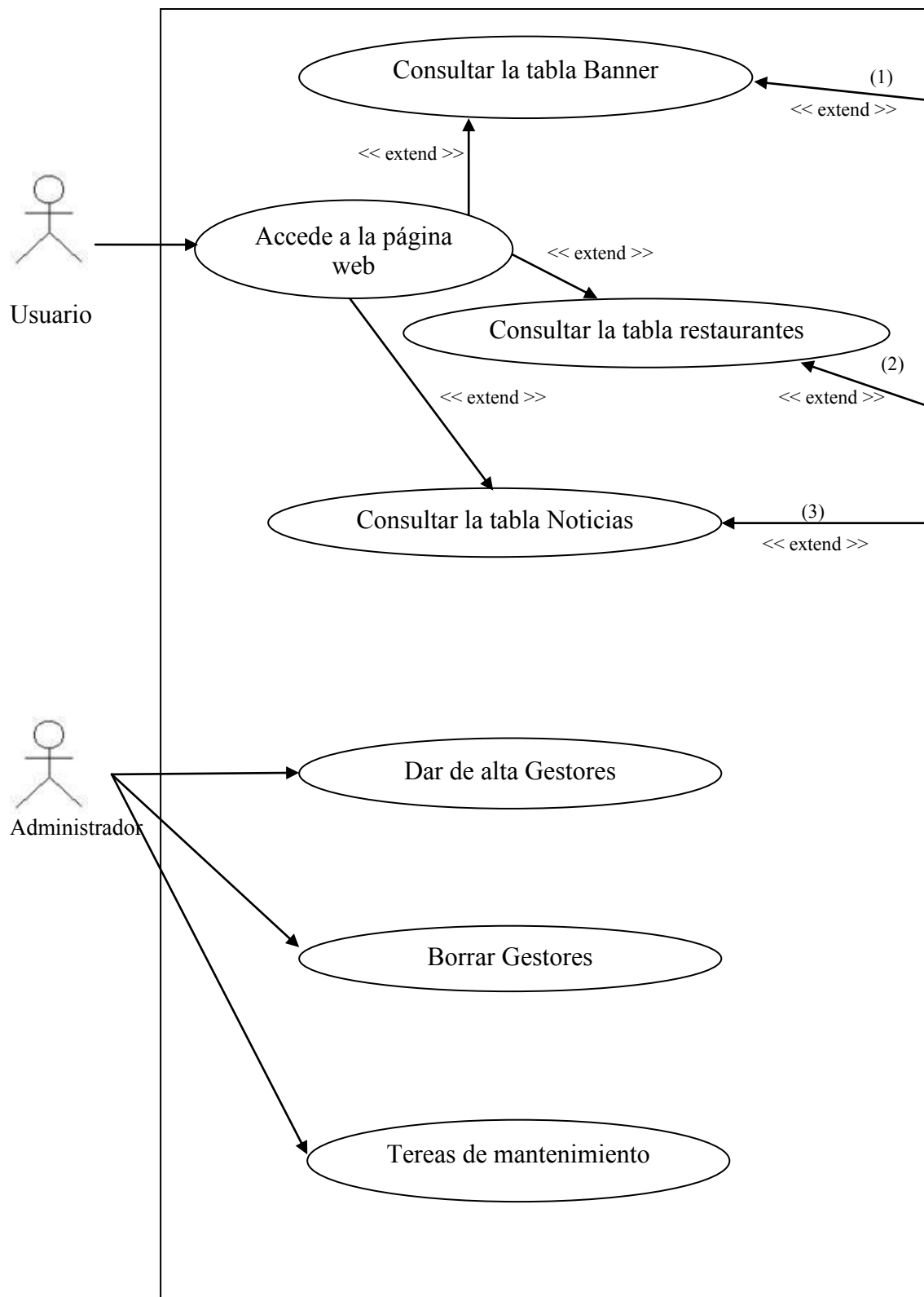


FIGURA 11 Casos de uso parte 1

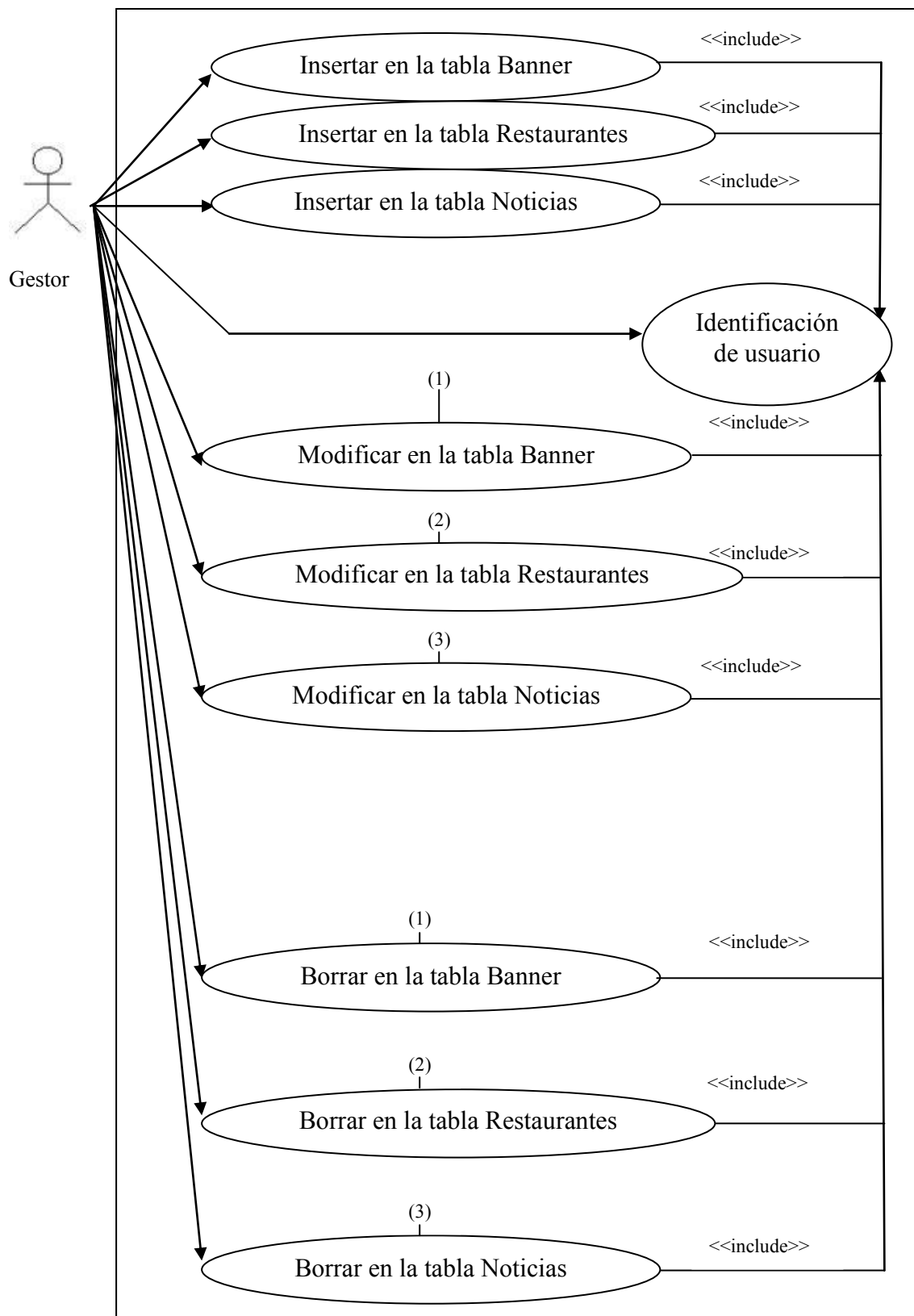


FIGURA 12 Casos de uso parte 2

Nota: Los números entre paréntesis representan relaciones entre los casos de uso, en donde se representa la relación de los casos de uso; modificar Diapositiva, modificar Restaurantes, modificar Noticias, borrar Diapositiva, borrar Restaurante y borrar Noticia.

Todas ellas tienen una relación <<extended>> con las consultas a las tablas donde se va a modificar o a borrar un registro. Estas relaciones son necesarias ya que para poder modificar un registro de una tabla es necesario visualizar en el gestor los datos de dicho registro, para ello se requiere realizar una consulta en la tabla de los datos de ese registro.

Lo mismo ocurre al borrar un registro de las tablas, es necesario ver en el gestor algún dato que diferencie a los registros de la tabla y así poder borrar de manera segura el registro.

A la hora de realizar algún caso de uso por el gestor, se requiere que el usuario se identifique en el sistema mediante un email y una contraseña. Es imprescindible que el usuario Gestor se identifique para poder acceder al gestor de la página web, en donde puede realizar todas las tareas asignadas anteriormente al usuario gestor.

2.4.4.- Especificación textual de los casos de uso

Se va a realizar una explicación textual de los casos de uso del sistema. Los casos de uso se han clasificado por los actores que interactúan con el sistema.

Usuario

Nombre	Accede a la página web
Actor	Usuario
Objetivo	Visualizar la pagina web
Precondiciones	Disponer de un ordenador con conexión a internet y un explorador de internet instalado.
Poscondiciones	Ninguna
Escenario básico	Un usuario accede a la página web para visualizarla. De manera transparente accede a la base de datos para recopilar información necesaria de algunas secciones, para ello se realizan consultas en la base de datos, se recupera la información necesaria y se muestra al usuario la pagina web con todo el contenido.

Tabla 32 Usuario accede a la página web

Administrador

Nombre	Dar de alta Gestores
Actor	Administrador
Objetivo	Dar de alta a un usuario gestor en el sistema
Precondiciones	Disponer de un email y una contraseña del usuario a dar de alta.
Poscondiciones	Nuevo usuario Gestor ingresado en el sistema.
Escenario básico	Un administrador accede a la base de datos e introduce en la tabla usuarios un nuevo registro, con un email del usuario y una contraseña. La contraseña puede ser dada por el administrador de manera automática o la puede dar el propio usuario que va a ser dado de alta.

Tabla 33 Dar de alta Gestores

Nombre	Borrar Gestores
Actor	Administrador
Objetivo	Eliminar a un usuario gestor del sistema
Precondiciones	Disponer del email del usuario a eliminar
Poscondiciones	Usuario Gestor eliminado del sistema.
Escenario básico	El administrador ingresa en la base de datos, realiza una búsqueda por email en la tabla usuarios, localiza al usuario gestor que se desea eliminar y se borra su registro de la tabla usuarios.

Tabla 34 Borrar Gestores

Nombre	Tareas de mantenimiento
Actor	Administrador
Objetivo	Realizar tareas de mantenimiento de la base de datos
Precondiciones	Problema en la base de datos o se requiere realizar un backup.
Poscondiciones	El problema ha desaparecido o se ha realizado el backup de la base de datos.
Escenario básico	<p>El administrador ingresa en la base de datos busca el problema, una vez localizado el problema lo soluciona.</p> <p>En caso de tener que realizar un backup de la base de datos, el administrador ingresa en la base de datos, realiza el backup y lo almacena.</p>

Tabla 35 Tareas de mantenimiento

Gestor

Nombre	Insertar en la tabla Banner
Actor	Gestor
Objetivo	Insertar un nuevo registro en la tabla Banner
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Nuevo registro en la tabla Banner.
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección banner, pincha en el enlace “nueva diapositiva” e introduce correctamente todos los datos solicitados.

Tabla 36 Insertar en la tabla Banner

Nombre	Insertar en la tabla Restaurantes
Actor	Gestor
Objetivo	Insertar un nuevo registro en la tabla Restaurantes
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Nuevo registro en la tabla Restaurantes.
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección restaurantes, pincha en el enlace nuevo restaurante e introduce correctamente todos los datos solicitados.

Tabla 37 Insertar en la tabla Restaurantes

Nombre	Insertar en la tabla Noticias
Actor	Gestor
Objetivo	Insertar un nuevo registro en la tabla Noticias.
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Nuevo registro en la tabla Noticias
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección noticias, pincha en el enlace nueva noticias e introduce correctamente todos los datos solicitados.

Tabla 38 Insertar en la tabla Noticias

Nombre	Modificar en la tabla Banner
Actor	Gestor
Objetivo	Modificar un registro de la tabla Banner
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Datos modificados del registro en la tabla Banner
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección banner, pincha en el enlace editar de la fila correspondiente al registro a modificar y modifica los datos que se requieren modificar.

Tabla 39 Modificar en la tabla Banner

Nombre	Modificar en la tabla Restaurantes
Actor	Gestor
Objetivo	Modificar un registro de la tabla Restaurantes
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Datos modificados del registro en la tabla Restaurantes
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección restaurantes, pincha en el enlace editar de la fila correspondiente al registro a modificar y modifica los datos que se requieren modificar.

Tabla 41 Modificar en la tabla Restaurantes

Nombre	Modificar en la tabla Noticias
Actor	Gestor
Objetivo	Modificar un registro de la tabla Noticias
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Datos modificados del registro en la tabla Noticias
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección noticias, pincha en el enlace editar de la fila correspondiente al registro a modificar y modifica los datos que se requieren modificar.

Tabla 40 Modificar en la tabla Noticias

Nombre	Borrar en la tabla Banner
Actor	Gestor
Objetivo	Borrar un registro de la tabla Banner
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Registro de la tabla Banner eliminado del sistema
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección banner, pincha en el enlace borrar de la fila correspondiente al registro a borrar y lo elimina.

Tabla 42 Borrar en la tabla Banner

Nombre	Borrar en la tabla Restaurantes
Actor	Gestor
Objetivo	Borrar un registro de la tabla Restaurantes
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Registro de la tabla Restaurantes eliminado del sistema
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección restaurantes, pincha en el enlace borrar de la fila correspondiente al registro a borrar y lo elimina.

Tabla 43 Borrar en la tabla Restaurantes

Nombre	Borrar en la tabla Noticias
Actor	Gestor
Objetivo	Borrar un registro de la tabla Noticias
Precondiciones	Identificarse como usuario gestor en el gestor.
Poscondiciones	Registro de la tabla Noticias eliminado del sistema
Escenario básico	El usuario Gestor se identifica en el gestor, accede a la sección noticias, pincha en el enlace borrar de la fila correspondiente al registro a borrar y lo elimina del sistema.

Tabla 44 Borrar en la tabla Noticias

2.5.- Características de los usuarios

2.5.1.- Introducción

El sistema deberá ofrecer una interfaz de usuario intuitiva, basada en el fácil acceso al contenido desde cualquier parte de la web y sencillo de manejar, para que no provoque problema alguno a los usuarios del mismo. Ha de ser soportado por todos los exploradores actuales.

2.5.2.- Restricciones

- *Restricciones de la política de la empresa:* la empresa solicita de manera explícita que la página web no solo sea funcional en los exploradores actuales, si no que también es necesario que la página web sea soportada por los exploradores de smartphones y tabletas.
- *Restricciones de privacidad:* El sistema deberá proteger los datos del sistema, tanto los datos de las secciones como los datos de los usuarios dados de alta como administradores en el sistema, asegurando que nadie pueda hacer un mal uso de los mismos.
- *Restricciones de fiabilidad:* El sistema deberá almacenar correctamente toda la información de manera coherente, evitando problemas como la redundancia o que no estén los datos debidamente actualizados en todo momento.
- *Restricciones de seguridad:* El sistema solo tiene un nivel de seguridad para los usuarios, ya que los únicos usuarios que pueden acceder al sistema son los usuarios gestores que pueden acceder al gestor. La seguridad ha de ser robusta, impidiendo cualquier tipo de infiltración de información.
Para ello es imprescindible que el servidor tenga la última versión de php, MySql y del servidor apache instalado.
- *Otras restricciones:* El sistema deberá proporcionar accesos rápidos y eficientes a la información en él contenida.

2.5.3.- Suposiciones y Dependencias

SUPOSICIONES

Se supone que todos los requisitos que han sido solicitados serán establecidos una vez aprobados por el cliente. Aunque por supuesto pueden modificarse a lo largo del tiempo, siempre y cuando sea aprobado por todas las partes y gestionado por el grupo desarrollador. En ese caso se realizará la modificación del sistema.

DEPENDENCIAS

Tanto la pagina web como el gestor contiene sentencias php que se comunican con la base de datos, dando las siguientes dependencias; la pagina web y el gestor se comunican con la base de datos a través de las sentencias php que son gestionadas por el servidor el cual accede a la base de datos, realiza las operaciones requeridas y devuelve el resultado al gestor.

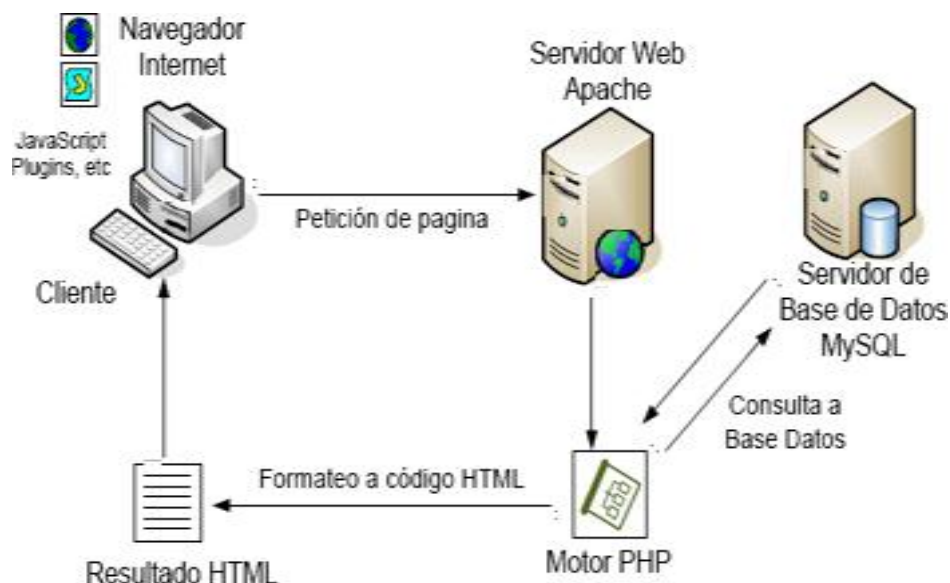


FIGURA 13 Dependencias

Por lo tanto las dependencias son; ordenador de un usuario comunica con nuestro servidor, este realiza sentencias php, accede a la base de datos, realiza las operaciones pertinentes en la base de datos, obtiene un resultado, formatea el resultado en código html y manda la pagina web al ordenador del usuario.

2.5.4.- Requisitos Futuros

En un futuro es posible que el sistema amplíe su funcionalidad, y se implemente un sistema de pedidos, en donde el usuario pueda realizar un encargo a uno de los restaurantes y le lleven el pedido a casa o bien el mismo usuario vaya a recoger su pedido al restaurante donde lo ha solicitado.

También cabe la posibilidad de implementar un sistema que permita a los usuarios realizar la búsqueda de restaurantes mediante provincias, previamente indicadas en un mapa.

Es posible que a medida que pase el tiempo se vayan incrementando las necesidades del sistema, en ese momento se volverá a valorar la mejora de este en función de las necesidades que vaya solicitando el cliente.

2.5.5.- Requisitos Específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados.

Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de “estabilidad”: Dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

- **(Requisito 1)** Para usuario administrador habrán de ser almacenados y guardados en la base de datos, únicamente los siguientes campos: email, contraseña.
- **(Requisito 2)** Los establecimientos deben poder ser establecimientos abiertos o próxima apertura.
- **(Requisito 3)** Los establecimientos han de tener los siguientes campos; nombre, foto, dirección, ciudad, teléfono, legal apertura, orden, horario mañana, horario tarde y próxima apertura. Donde legal apertura ha de valer 0 o 1, indicando si el establecimiento requiere de un texto legal estándar indicando que los horarios son los marcados por el centro comercial. El campo próxima apertura ha de valer 0 o 1, indica si el establecimiento es próxima apertura o no.

- **(Requisito 4)** En caso de que se creé un establecimiento como próxima apertura los campos básicos que debe rellenar son; nombre y ciudad.
- **(Requisito 5)** Para crear una noticia es necesario rellenar todos los campos establecidos excepto el campo foto. Los campos son; título, fuente de la noticia, fecha, contenido, foto, link a la noticia.
- **(Requisito 6)** Para crear una diapositiva del banner es necesario rellenar todos los campos establecidos; contenido, posición del texto y la imagen del fondo.
- **(Requisito 7)** Todas las tablas de la base de datos tienen un campo id, siendo la clave primaria y con valor autoincremental, asignado por el sistema. Este campo es para tareas de mantenimiento y búsqueda en la base de datos.

2.5.6.- Requisitos de Interfaces Externas

INTERFACES DE USUARIO

La interfaz de usuario debe ser intuitiva, fácil de manejar y fácil de acceder a cualquier contenido desde cualquier apartado de la pagina web. La navegación por la aplicación podrá realizarse exclusivamente con el ratón, conjuntamente con el teclado o no (a través de pantallas táctiles).

INTERFACES HARDWARE

Se requiere un ordenador o smarphone con conexión a internet para poder ver la pagina web y un servidor donde esta albergada la pagina web y a la base de datos.

INTERFACES SOFTWARE

Se requiere un servidor con Apache instalado, php y MySql en su última versión, para garantizar la máxima seguridad.

INTERFACES DE COMUNICACIÓN

La interfaz de comunicación se realiza a través de internet, por lo tanto es necesario que el usuario tenga acceso a internet para poder visualizar la pagina web.

2.5.7.- Requisitos de Rendimiento

Se cuenta con que la frecuencia de uso del sistema, será diaria, en todo momento deberá mostrar un rendimiento óptimo.

2.5.8.- Restricciones del Diseño

El diseño final es una pagina web una navegación lateral, siendo la principal restricción de diseño del sistema

El ciclo de vida elegido para desarrollar el sistema será a través de prototipos, entregando al cliente diversos prototipos a lo largo del tiempo para ajustar al máximo el resultado final con las necesidades del cliente.

2.6.- Arquitectura de la base de datos

2.6.1.- Introducción

Se van a explicar porque se ha usado MySql server para el desarrollo de la base de datos, de que componentes esta formada y como interactúa con el portal web comercial desarrollado en este proyecto de fin de carrera.

2.6.2.- Arquitectura de la base de datos

Se ha optado por la utilización de MySql, como sistema de gestión de la base de datos.

El servidor de bases de datos relacionales MySQL es muy rápido, fiable y fácil de usar, ya que en cooperación con los usuarios desarrolladores de Open Source a nivel mundial MySQL Server se desarrolló originalmente para manejar grandes bases de datos más rápido que las soluciones de gestores de bases de datos existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años.

MySQL Server trabaja en entornos cliente/servidor o incrustados; el software de bases de datos MySQL es un sistema cliente/servidor que consiste en un servidor SQL que trabaja con diferentes programas y bibliotecas cliente, herramientas administrativas y diversas interfaces de programación para aplicaciones (APIs). También se proporciona el MySQL Server como biblioteca incrustada, que se puede incluir en una aplicación para obtener un producto más pequeño, rápido y fácil de administrar. En este caso se va a usar PHP para trabajar con el servidor SQL.

MySQL posee un buen control de acceso de usuarios y seguridad en los datos. Además tiene una integración perfecta con el lenguaje PHP.

MySql da soporte completo para cláusulas, funciones, tipos de datos y comandos estándar y extendidos del estándar SQL. También da soporte para control de transacciones en tablas transaccionales, y soporte para procedimientos almacenados, subconsultas y disparadores (Triggers) en las últimas versiones de MySQL (5.x).

Soporta múltiples modos asignados para comportarse como otros gestores de bases de datos, definiendo la validación o no de los valores erróneos o incorrectos y la forma en que se ingresan los datos.

Independientemente del almacén de datos utilizado, la aplicación o el servicio, se utilizan componentes lógicos para obtener acceso a los datos. Los componentes lógicos abstraen al usuario de la semántica del almacenamiento de datos y la tecnología de acceso a datos, proporcionando una interfaz simple de programación para la realización de operaciones y recuperación de datos.

Cada uno de estos componentes suele proporcionar métodos para realizar operaciones sobre una entidad determinada de la aplicación.

En este caso, la aplicación web contiene componentes lógicos de acceso a datos, por lo que es de gran utilidad utilizar un componente de ayuda de acceso a datos genéricos para administrar las conexiones de las bases de datos, ejecutar comandos y almacenar parámetros en caché, entre otros. Los componentes lógicos de acceso a datos proporcionan la lógica necesaria para obtener acceso a los datos almacenados, mientras que el componente de ayuda para el acceso a datos proporciona una API para el acceso a datos y la configuración de la conexión a éstos, permitiendo de esta forma la reducción de código duplicado. El componente de ayuda de acceso a datos bien diseñado no debe repercutir negativamente en el rendimiento y proporciona una optimización del acceso a datos.

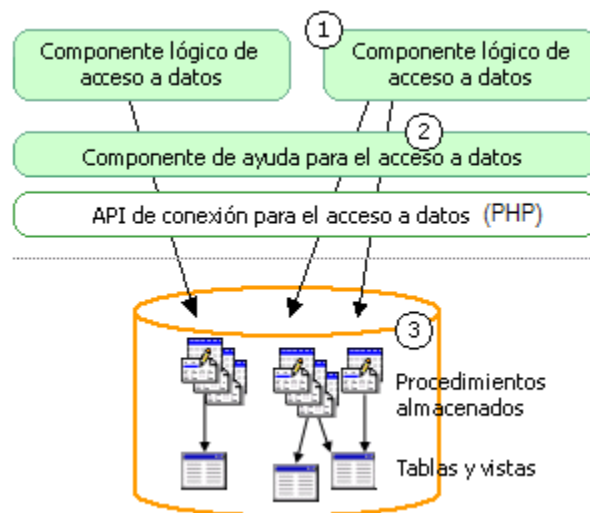


FIGURA 14 Componentes lógicos de acceso a datos

1. Los componentes lógicos de acceso a datos exponen métodos para insertar, eliminar, actualizar y recuperar datos, incluyendo la provisión de funcionalidad de paginación al recuperar grandes cantidades de datos.
2. Puede utilizar un componente de ayuda de acceso a datos para centralizar la administración de la conexión y todo el código relacionado con un origen de datos específico.
3. Se recomienda implementar las consultas y operaciones de datos como procedimientos almacenados (si es compatible con el origen de datos) para mejorar el rendimiento y la facilidad de mantenimiento.

El uso de componentes lógicos de acceso a datos se utiliza para todas el acceso a datos, tanto para el gestor como para la pagina web (consultas, inserciones, borrados, etc.).

Los componentes lógicos de acceso a datos proporcionan acceso simple a las funcionalidades de la base de datos (consultas y operaciones sobre los datos), devolviendo estructuras de datos simples y complejas. Ocultan la semántica de la invocación y el formato del almacenamiento de los datos de la página web y del gestor que las consumen. La implementación de una lógica propia de acceso a datos en los componentes lógicos de acceso a datos permite encapsular toda la lógica de acceso a datos de la aplicación completa en una única ubicación central, lo que facilita el mantenimiento y la extensibilidad de la aplicación.

Lo más recomendable es diseñar cada uno de los componentes lógicos de acceso a datos para trabajar con un único almacén de datos. En este caso se centra en la pagina web que realiza acciones de consulta sobre la base de datos y el gestor cuyas operaciones son mas complejas, además de consulta de datos, crea nuevos datos en las tablas, elimina datos y los modifica.

2.7.- Planificación

2.7.1.- Introducción

En este apartado se va a estimar un presupuesto y un tiempo para desarrollar la aplicación web de este proyecto de fin de carrera. Para poder realizar la estimación del presupuesto necesario para llevar a cabo el proyecto se tienen en cuenta las tareas a realizar para la creación del proyecto, así como el tiempo necesario hasta la finalización y las personas dedicadas al mismo.

Tal y como vimos en la asignatura de Ingeniería del Software, antes de empezar cualquier proyecto es necesario realizar la planificación y organización del mismo, para ello se ha utilizado el método COCOMO.

El método COCOMO se basa en el cálculo de números de líneas de código como medio de determinación del tamaño del proyecto. EL punto de partida los marcan unas especificaciones concretas, contenidas en la documentación del prototipo original, así como del tamaño del software desarrollado en dicho prototipo.

El ciclo de vida elegido para desarrollar el sistema es el ciclo de “Entrega Evolutiva”:

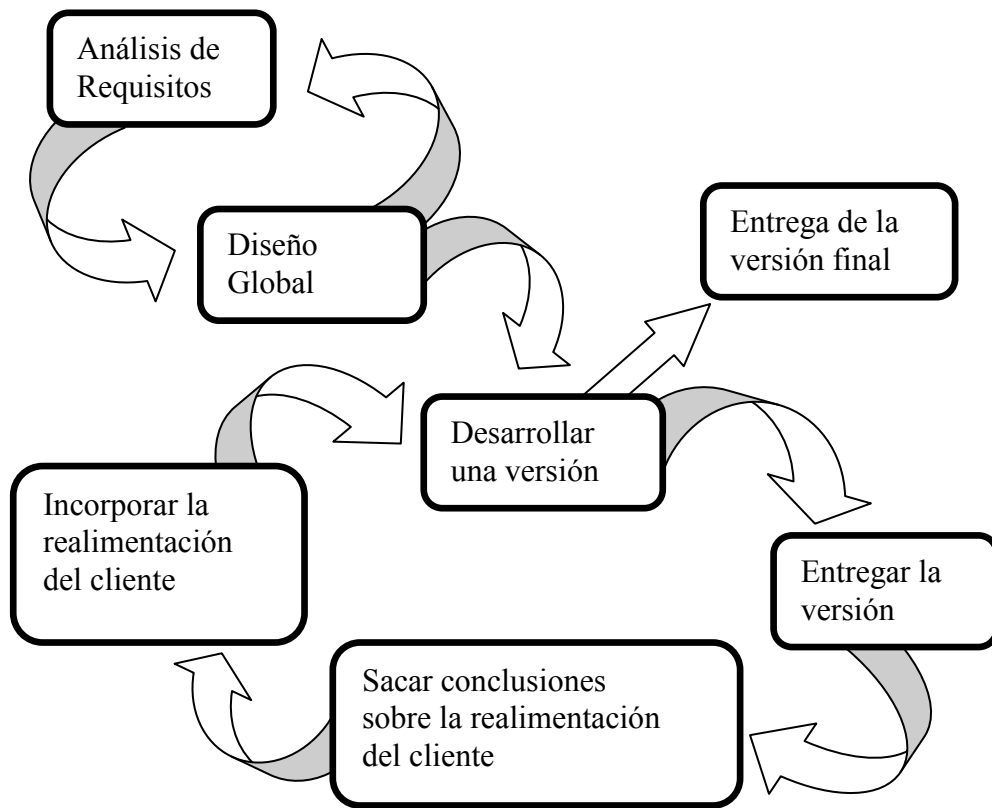


FIGURA 15 Ciclo de vida “Entrega Evolutiva”

Este tipo de ciclo de vida esta compuesto por las siguientes etapas:

- Análisis de Requisitos
- Diseño global
- Desarrollar una versión
- Entregar la versión
- Sacar conclusiones sobre la realimentación del cliente
- Incorporar la realimentación del cliente
- Entrega de la versión final

El ciclo de vida se basa en la entrega de versiones al cliente, lo que asegura una aplicación completamente personalizada, ya que es el propio cliente el que da las pautas para la corrección o mejora de la página web.

Para poder realizar un cálculo aproximado del coste del proyecto es necesario realizar este tipo de estimación, la cual nos dará de forma aproximada el coste y la duración del proyecto.

2.7.2.- Estimación

Para realizar la estimación vamos emplear el método COCOMO. Existen dos variantes muy extendidas de este método de estimación, que son COCOMO 81 y COCOMO II, siendo COCOMO II posterior a COCOMO 81, y por lo tanto COCOMO II ajusta mejor la estimación de costes y de esfuerzo a los proyectos software más actuales. Por esta razón vamos a emplear en la estimación de costes y tiempo el método COCOMO II y no COCOMO 81.

2.7.3.- Estimación de los Puntos de Función

COCOMO II realiza la estimación de coste y duración basándose en las líneas de código. Para poder calcular las líneas de código es necesario calcular los puntos de función. Para calcular los puntos de función se deben identificar los siguientes tipos de ítems:

- Entradas Externas (Inputs): Entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno.
- Salidas Externas (Outputs): Salida de datos de usuario o de control que deja el límite del sistema de software.
- Consultas Externas (Queries): Combinación única de entrada-salida, donde una entrada causa y genera una salida inmediata, como un tipo de solicitud externa.
- Archivos Externos de Interfase (Interfases): Archivos transferidos o compartidos entre sistemas de software.
- Archivo Lógicos Internos (Archivos): Incluye cada archivo lógico, es decir cada grupo lógico de datos que es generado, usado, o mantenido por el sistema de software.

El alta, modificación y borrado de un usuario gestor no se contempla en esta estimación por no estar incluidas dichas acciones en la aplicación software a desarrollar. Estas acciones son realizadas por administradores de la base de datos. Este sistema de alta de usuarios gestores es por petición del cliente.

A continuación se muestran las entradas externas, salidas externas, consultas externas, archivos externos de Interfase y archivo lógicos internos:

Entradas externas (Inputs)

Crear diapositiva para el banner
Crear restaurante
Crear noticia
Modificar diapositiva para el banner
Modificar restaurante
Modificar noticia
Modificar tiempo Banner
Borrar diapositiva para el banner
Borrar restaurante
Borrar noticia

Salidas externas (Outputs)

Envío email de consulta de clientes
Mensajes de error (15)

Consultas Externas (Queries)

Detalles usuario gestor (inicio de sesión)
Detalles diapositiva para el banner
Destalles restaurante
Detalles noticia
Tiempo banner

Archivos Externos de Interfase (Interfases)

Fotos

Archivo Lógicos Internos (Archivos)

Base de datos

Se van a clasificar los ítems obtenidos en función de su complejidad en la siguiente tabla:

Origen	Baja	Media	Alta
Entradas externas (Inputs)		Crear diapositiva para el banner Crear restaurante Crear noticia Modificar diapositiva para el banner Modificar restaurante Modificar noticia Modificar tiempo Banner Borrar diapositiva para el banner Borrar restaurante Borrar noticia	
Salidas externas (Outputs)	Mensajes de error (15)	Envió email de consulta de clientes	
Consultas Externas (Queries)	Inicio de sesión	Detalles diapositiva para el banner Destalles restaurante Detalles noticia Tiempo banner	
Archivos Externos de Interfase (Interfases)	Fotos		
Archivo Lógicos Internos (Archivos)		Base de datos	

Tabla 45 Ítems obtenidos

A continuación se muestran los valores para cada clasificación:

	Baja	Media	Alta
Entradas externas (Inputs)	3	4	6
Salidas externas (Outputs)	4	5	7
Consultas Externas (Queries)	3	4	6
Archivos Externos de Interfase (Interfases)	5	7	10
Archivo Lógicos Internos (Archivos)	7	10	15

Tabla 46 Valores de ítems

En la siguiente tabla se calcula el total de puntos de función sin ajustar obtenidos para la clasificación anterior:

	Baja			Media			Alta		
	Nº	Multiplicador	Total	Nº	Multiplicador	Total	Nº	Multiplicador	Total
Entradas externas (Inputs)	0	3	0	10	4	40	0	6	0
Salidas externas (Outputs)	15	4	60	1	5	5	0	7	0
Consultas Externas (Queries)	1	3	3	4	4	16	0	6	0
Archivos Externos de Interfase (Interfases)	1	5	5	0	7	0	0	10	0
Archivo Lógicos Internos (Archivos)	0	7	0	1	10	10	0	15	0
Totales	68			71			0		
Total	139								

Tabla 47 Puntos de función sin ajustar

Para calcular los puntos de función ajustados se requiere del cálculo de la variable “Factor de Complejidad Técnica (TCF)”. Esta variable es necesaria para aplicar la siguiente ecuación, de la que se obtienen los puntos de función ajustados:

$$TCF = 0.65 + 0.01 \times \sum_{i=1}^{14} F_i$$

Influencia en el proyecto:

- 0: Sin influencia
- 1: Incidental
- 2: Moderado
- 3: Medio
- 4: Significativo
- 5: Esencial

Factor	0	1	2	3	4	5
F1: Mecanismos de recuperación y back-up confiables						X
F2: Comunicación de Datos					X	
F3: Funciones de Procesamiento Distribuido				X		
F4: Performance ¿Es crítico el rendimiento?						X
F5: Configuración usada rigurosamente						X
F6: Entrada de datos on-line						X
F7: Factibilidad Operativa				X		
F8: Actualización de archivos on-line				X		
F9: Interfaces complejas				X		
F10: Procesamiento interno complejo				X		
F11: Reusabilidad de código					X	
F12: Fácil Instalación		X				
F13: Soporte de múltiples instalaciones					X	
F14: Facilidad de cambios y amigabilidad						X
Sumatorio	0	1	0	5	3	5
Sumatorio x Multiplicador		1		15	12	25
Total	53					

Tabla 48 Valores del factor de complejidad

$$\begin{aligned} \text{TCF} &= 0.65 + 0.01 \times \sum_{i=1}^{14} F_i \\ \text{TCF} &= 0.65 + 0.01 \times 53 \\ \text{TCF} &= 1.18 \end{aligned}$$

El cálculo resultante para los puntos de función es:

$$\text{PFA} = 139 \times 1,18 = 164,02$$

2.7.4.- Estimación de las líneas de código

Para calcular las líneas de código, primero hemos de multiplicar los puntos de función obtenidos por el promedio de líneas de código por punto de función según el lenguaje empleado. Para ello se va a emplear la tabla creada por “Quantitative Software Management”, la cual se puede encontrar en la dirección de su página web <http://www.qsm.com/resources/function-point-languages-table>. En esta tabla encontramos el número de líneas de código promedio para los puntos de función de varios lenguajes de programación, en donde se encuentra Web Scripts con un valor medio igual a 15.

$$\begin{aligned} \text{SLCO} &= 164,02 \times 15 = 2460,3 \\ \text{KSLOC} &= 2460,3 / 1000 = 2,4603 \end{aligned}$$

2.7.5.- Estimación del esfuerzo, duración del proyecto y personal necesario

Para calcular el resto de variables COCOMO dispone de varios modelos, en este caso la arquitectura del proyecto está completamente definida, la creación de páginas web no es algo nuevo y ya se ha realizado el análisis del sistema, por lo tanto se va a emplear el modelo Post-Arquitectura, cuya función para calcular el esfuerzo es:

$$\begin{aligned} \text{PM} &= a \times \text{Size} \times \prod_{i=1}^b \text{EM}_i \\ i &= 1 \text{ a } 16 \end{aligned}$$

El esfuerzo nominal para este modelo se ajusta usando 17 factores multiplicadores de esfuerzo.

$$b = 1.01 + 0.01 \times \sum \text{SF}_j \quad (j = 1 \text{ a } 5)$$

En las siguientes tablas se muestran los valores para los factores de escala y que criterios seguir par la clasificación de cada uno, junto con los valores asignados según su clasificación. Los valores elegidos para cada factor se muestran sombreados de color gris en la tabla:

Factor de Escala Wj	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
Precedencia PREC	Completamente sin precedentes	Ampliamente sin precedentes	Algún Precedente	Generalmente Familiar	Ampliamente Familiar	Completamente Familiar
Flexibilidad en el desarrollo FLEX	Rigurosa	Relajación Ocasional	Alguna Relajación	Conformidad en General	Alguna Conformidad	Metas Generales
Arquitectura/ Resolución de riesgo RESL	Poca (20%)	Alguna (40%)	Siempre (60%)	Generalmente 75%)	Principalmente (90%)	Completo (100%)
Cohesión de equipo TEAM	Interacciones Dificiles	Interacciones con alguna dificultad	Interacciones básicamente cooperativas	Ampliamente Cooperativas	Altamente Cooperativas	Interacciones Sin Fisuras
Madurez del proceso PMAT	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto

Tabla 49 Factores de escala

Tabla de valores:

Factor de Escala Wj	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
Precedencia PREC	6,20	4,96	3,72	2,48	1,24	0
Flexibilidad en el desarrollo FLEX	5,07	4,05	3,04	2,03	1,01	0
Arquitectura/Resolución de riesgo RESL	7,07	5,65	4,24	2,83	1,41	0
Cohesión de equipo TEAM	5,48	4,38	3,29	2,19	1,10	0
Madurez del proceso PMAT	7,80	6,24	4,68	3,12	1,56	0

Tabla 50 Valores factores de escala

$$b = 1.01 + 0.01 \times (1,24 + 4,05 + 7,07 + 0 + 4,68) = 1.01 + 0.01 \times 17,4$$

$$b = 1,1804$$

Una vez calculado “b” se calcula Π Emi empleado las siguientes tablas, en donde se indica que valor otorgar a los 17 factores que componen dicha variable:

	Factor	Muy Bajo	Bajo	Normal	Alto	Muy Alto	Extra
Producto	RELY	Inconvenientes insignificantes, que afectan solamente a los desarrolladores	Mínimas pérdidas al usuario, fácilmente recuperables	Pérdidas moderadas al usuario recuperables sin grandes inconvenientes	Pérdida financiera elevada o inconveniente humano masivo	Vida humana en riesgo	
	DATA		DB bytes/Pgm SLOC <10	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P > 1000$	
	CPLX	Ver tabla Factor Multiplicador CPLX. Complejidad del Producto.					
	RUSE		Ningún componente reusable	Reusable dentro del mismo proyecto	Reusable dentro de un mismo programa	Reusable dentro de una misma línea de productos	Reusable dentro de múltiples líneas de producto
	DOCU	Muchas necesidades del ciclo de vida sin cubrir	Algunas necesidades del ciclo de vida sin cubrir	Necesidades del ciclo de vida cubiertas en su justa medida	Necesidades del ciclo de vida cubiertas ampliamente	Necesidades del ciclo de vida cubiertas excesivamente	
Plataforma	TIME			Uso de $\leq 50\%$ del tiempo de ejecución disponible	70%	85%	95%
	STOR			Uso de $\leq 50\%$ del porcentaje total de almacenamiento	70%	85%	95%
	PVOL		Un cambio principal cada 12 meses. Un cambio menor todos los meses	Cambio principal cada 6 meses. Cambio menor cada 2 semanas	Cambio principal cada 2 meses. Cambio menor uno por semana	Cambio principal cada 2 semanas. Cambio menor cada 2 días	
Personal	ACAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil	
	PCAP	15 percentil	35 percentil	55 percentil	75 percentil	90 percentil	
	PCON	48 % por año	24 % por año	12 % por año	6% por año	3 % por año	
	AEXP	≤ 2 meses	≤ 6 meses	1 año	3 años	6 años	
	PEXP	≤ 2 meses	≤ 6 meses	1 año	3 años	6 años	
	LTEX	≤ 2 meses	≤ 6 meses	1 año	3 años	6 años	
Proyecto	TOOL	Herramientas que permiten editar, codificar, depurar	Herramientas simples con escasa integración al proceso de desarrollo	Herramientas básicas, integradas moderadamente	Herramientas robustas y maduras, integradas moderadamente	Herramientas altamente integradas a los procesos, métodos y reuso	
	SITE Ubicación Espacial	Internacional	Multi-ciudad y multi-compañía	Multi-ciudad o multi-compañía	Misma ciudad o área metropolitana	Mismo Edificio o complejo	Completamente Centralizado
	SITE Comunicación	Algún teléfono, mail	Teléfonos individuales, FAX	Email de banda angosta	Comunicaciones electrónicas de banda ancha	Comunicaciones electrónicas de banda ancha, ocasionalmente videoconferencia	Multimedia Interactiva
	SCED	75% del nominal	85% del nominal	100% del nominal	130% del nominal	160% del nominal	

FIGURA 16 Factores de costo Modelo Post-Arquitectura. [Boehm 1995/1] [Boehm 1995/2]

	Operaciones de Control	Operaciones computacionales	Operaciones dependientes de los dispositivos	Operaciones de administración de datos	Operaciones de administración de interfaces de usuario
Muy Bajo	Pocas estructuras sin anidamiento: DO, CASE, IF_THEN_ELSE. Composición modular simple por medio de llamadas a procedimientos o simples script	Evaluación de una expresión simple Por ejemplo: $A=B+C*(D-E)$	Sentencias de lectura / escritura con formatos simples	Arreglos simples en memoria principal. Consultas, actualizaciones a COTS-DB	Generadores de reportes, Formularios de entrada simples.
Bajo	Estructuras anidadas sencillas	Evaluación de expresiones de complejidad moderada Por ejemplo: $D=\text{SQRT}(B**2-4.*A*C)$	Ninguna necesidad de dispositivos especiales para procesamiento de I/O	Archivo que subsiste sin cambios de estructuras de datos, ni ediciones ni archivos intermedios. Consultas y actualizaciones a COTS-DB moderadamente complejas	Uso de generadores de interfaces de usuario gráficas simples
Nominal	Uso mayoritario de anidamientos sencillos . Algunos controles entre módulos. Tablas de decisión. Pasaje de mensajes o llamadas a subrutinas. Soporte para procesamiento distribuido	Uso de rutinas estándares de matemática y estadística Operaciones básicas con matrices y vectores	Procesamiento de Entradas/Salidas que incluye selección de dispositivo, procesamiento de errores y chequeo de estado	Varios archivos de entrada y solo un archivo de salida. Cambios estructurales sencillos y ediciones simples. Consultas y actualizaciones a COTS-DB complejas	Uso simple de algunos dispositivos
Alto	Programación estructurada con alto grado de anidamiento con predicados compuestos. Control de cola y pila. Procesamiento distribuido. Control en tiempo real con un procesador	Análisis numérico básico: Interpolación, ecuaciones diferenciales ordinarias, redondeos, truncamientos	Operaciones de Entrada/salida a nivel físico (traducciones a direcciones de almacenamiento físico, seeks, read, etc.) Optimización de superposición Entradas/Salidas	Triggers simples activados por flujos de datos. Reestructuración compleja de datos	Uso de un conjunto de dispositivo de Multimedia, Entrada/Salida de Procesamiento de voz
Muy Alto	Codificación recursiva. Manejo de interrupciones con prioridad fija. Sincronización de tareas, complejas llamadas a subrutinas. Procesamiento distribuido heterogéneo. Control en tiempo real con un procesador	Análisis numérico estructurado: Matrices de ecuaciones. Ecuaciones diferenciales parciales	Rutinas para el control de interrupciones, enmascaramiento. Manejo de líneas de comunicación	Coordinación de base de datos distribuidas Disparadores complejos. Optimización de búsqueda	Multimedia, Gráficos dinámicos, Gráficos 2D y 3D de moderada complejidad
Extra Alto	Planificación múltiple de recursos con cambio dinámico de prioridades. Control al nivel de microcódigo. Control en tiempo real distribuido	Análisis numérico no estructurado: Datos estocásticos. Análisis de ruido con alto grado de precisión	Codificación de dispositivos dependientes del tiempo Operaciones microprogramadas. Performance crítica con relación a I/O	Alto grado de acoplamiento, relaciones dinámicas, estructuras de objeto, administración de datos en lenguaje natural	Multimedia compleja Realidad virtual

FIGURA 17 Factor Multiplicador CLPX. Complejidad del producto

Seleccionamos los valores de cada factor según las tablas anteriores:

RELY (Confiabilidad requerida): Muy bajo

DATA (Tamaño de la base de datos): Alto, la base de datos contendrá aproximadamente 150 líneas de código.

CPLX (Complejidad del producto): Normal

RUSE (Requerimientos de reusabilidad): Muy alto, parte del código podrá ser usado para la creación de otras paginas web (por ejemplo func.php).

DOCU (Documentación acorde a las diferentes etapas del ciclo de vida): Normal

TIME (Restricción del tiempo de ejecución): Normal

STOR (Restricción del almacenamiento principal): Alto

PVOL (Volatilidad de la plataforma): Normal

ACAP (Capacidad del analista): Muy alto

PCAP (Capacidad del programador): Muy alto

PCON (Continuidad del personal): Normal

AEXP (Experiencia en la aplicación): Normal

PEXP (Experiencia en la plataforma): Normal

LTEX (Experiencia en el lenguaje y las herramientas): Normal

TOOL (Uso de herramientas de software): Normal

SITE (Desarrollo multisitio): Alto

SCED (Cronograma requerido para el desarrollo): Normal

En la siguiente tabla se muestran el valor de cada factor, se marca con fondo gris el valor seleccionado anteriormente para cada factor:

Factor	Muy Bajo	Bajo	Normal	Alto	Muy alto	Extra
RELY	0,82	0,92	1,00	1,10	1,26	XXXX
DATA	XXXX	0,90	1,00	1,14	1,28	XXXX
CPLX	0,73	0,67	1,00	1,17	1,34	1,74
RUSE	XXXX	0,95	1,00	1,07	1,15	1,24
DOCU	0,81	0,91	1,00	1,11	1,23	XXXX
TIME	XXXX	XXXX	1,00	1,11	1,29	1,63
STOR	XXXX	XXXX	1,00	1,05	1,17	1,46
PVOL	XXXX	0,87	1,00	1,15	1,30	XXXX
ACAP	1,42	1,19	1,00	0,85	0,71	XXXX
PCAP	1,34	1,15	1,00	0,88	0,76	XXXX
PCON	1,29	1,12	1,00	0,90	0,81	XXXX
AEXP	1,22	1,10	1,00	0,88	0,81	XXXX
PEXP	1,19	1,09	1,00	0,91	0,85	XXXX
LTEX	1,20	1,09	1,00	0,91	0,84	XXXX
TOOL	1,17	1,09	1,00	0,90	0,78	XXXX
SITE	1,22	1,09	1,00	0,93	0,86	0,80
SCED	1,43	1,14	1,00	1,00	1,00	XXXX

Tabla 51 Valor de los Factores de costo

$\Pi E_{mi} = 0,82 \times 1,14 \times 1,00 \times 1,15 \times 1,00 \times 1,00 \times 1,05 \times 1,00 \times 0,71 \times 0,76 \times 1,00 \times 1,00 \times 1,00 \times 1,00 \times 0,93 \times 1,00$

$\Pi E_{mi} = 0,566448893388$

$PM = a \times Size \times \Pi E_{mi}$
 $i = 1 \text{ a } 16$

Sabiendo que “a” es la constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño y que tiene un valor de $a=2.94$, se calcula:

$PM = 2,94 \times 2,4603 \times 0,566448893388$
 $PM = 4,8198150888201930150089109327786 \approx 4.82 \text{ personas-mes}$

2.7.6.- Estimación del tiempo y coste del proyecto

Se pasa a calcular el tiempo de duración del desarrollo del proyecto, para ello se emplea la siguiente formula:

$$TDEV = (3.0 \times PM^{(0.33 + 0,2 \times (B - 1,01))}) \times SCED\% / 100$$

En los cálculos anteriores se hayo el valor de B:

$$B = 1,1804$$

Y SCED% es valor del factor de esfuerzo que representa el cronograma requerido para el desarrollo. Se le dio un valor normal, cuyo valor es 1, y en porcentaje es igual al 100%. Aplicando el algoritmo anterior se obtiene el siguiente valor para TDEV:

$$TDEV = (3.0 \times 4.82^{(0.33 + 0,2 \times (1,1804 - 1,01))}) \times 1^{0,36408} = 3.0 \times 4.82$$

$$TDEV = 5,3186941909552022007661379151694 \approx 5,32 \text{ meses} \approx 5 \text{ meses}$$

Pm es igual a personas mes, por lo tanto para calcular as personas necesarias se emplea la siguiente ecuación:

$$PM = \text{Personas} * TDEV$$

$$\text{Personas} = PM / TDEV$$

$$\text{Personas} = 4.82 / 5,32 = 0,90601503759398496240601503759398 \approx 0.9 \approx 1$$

Se necesita de una persona para realizar el proyecto. Para calcular el coste se necesita el coste mensual y la herramienta software COCOMO II. Para este proyecto el coste mensual será igual a 1500€/mes. Aplicando los valores obtenidos en las tablas anteriores en la herramienta software COCOMO II se obtiene el siguiente resultado:

Project Name: **Sushi Chung** Scale Factor Schedule

Development Model: **Post Architecture** ▼

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Sushi Chung	S:2460	1500.00	0.57	Non-Specified	7.9	4.5	548.9	6722.19	2.7	0.8	0.0

	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Total Lines of Code: 2460	Optimistic	3.6	5.5	686.2	5377.75	2.2	0.7	
	Most Likely	4.5	5.9	548.9	6722.19	2.7	0.8	0.0
	Pessimistic	5.6	6.3	439.1	8402.74	3.4	0.9	

FIGURA 18 COCOMO II

Se aplican en el software COCOCOMO II los valores obtenidos anteriormente:

base + Incr % = rating

Product: RELY DATA DOCU CPLX RUSE

base VLO HI NOM NOM VHI

Incr% 0% 0% 0% 0% 0%

Platform: TIME STOR PVOL

base NOM HI NOM

Incr% 0% 0% 0%

Personnel: ACAP PCAP PCON APEX LTEX PLEX

base VHI VHI NOM NOM NOM NOM

Incr% 0% 0% 0% 0% 0% 0%

Project: TOOL SITE

base NOM HI

Incr% 0% 0%

User: USR1 USR2

base NOM NOM

Incr% 0% 0%

EAF is also affected by Schedule

EAF: **0.57**

FIGURA 19 Factores de costo Modelo Post-Arquitectura en COCOMO II

COCOMO II da un coste total aproximado de **6722,19 € para 5 meses**. Las diferencias entre el tiempo obtenido y el tiempo obtenido por la herramienta software COCOMO II se deben a las aproximaciones de los valores obtenidos para las diferentes variables.

Para comprobar que se han realizado los cálculos correctamente se observa el valor de EAT, en el calculo manual esta variable se denomina Π_{Emi} , dando el mismo resultado, donde la herramienta COCOMO II realiza una aproximación del valor obtenido, mientras que en los cálculos realizados anteriormente no se realiza dicha aproximación para obtener mayor exactitud en los cálculos.

2.7.7.- Estimación del tiempo y coste del proyecto por fases.

En la siguiente tabla se muestra la duración y el coste por fases, para ello se determina que un mes equivale a 100 horas de trabajo, dando un total de $5 \cdot 100 = 500$ horas:

	<i>COSTE%</i>	<i>DURACIÓN%</i>	<i>COSTE FASE</i>	<i>DURACIÓN FASE</i>
ANÁLISIS	0.10	0.05	672,219€	25 HORAS
DISEÑO	0.35	0.35	2352,7665 €	175 HORAS
PROGRAMACIÓN	0.40	0.55	2688,876 €	275 HORAS
PRUEBAS	0.10	0.05	672,219 €	50 HORAS
IMPLANTACIÓN	0.05	0.03	336,1095 €	15 HORAS

Tabla 52 Duración y coste estimados, por fases

Hay que tener en cuenta que toda página web requiere de un alojamiento en un servidor y un dominio para acceder a dicha pagina web.

Por lo tanto al presupuesto hay que añadirle el precio por alojamiento, al que generalmente va asociado un servicio de mantenimiento. El servicio de alojamiento y mantenimiento se cobra por meses.

En cuanto al registro del dominio, no se va a presupuestar debido a su bajo coste (5€ por año aproximadamente).

Alojamiento + mantenimiento = 50€/mes

Presupuesto final = 6722,19 € + 50€/mes

2.8.- Gráfico Gantt

2.8.1.- Introducción

El Ciclo de Vida elegido es de “Entrega Evolutiva”, dividiéndose el proyecto en las siguientes fases:

- a) Análisis
- b) Diseño
- c) Desarrollar una versión
- d) Entregar la versión
- e) Sacar conclusiones sobre la realimentación del cliente
- f) Incorporar la realimentación del cliente
- g) Desarrollar una versión
- h) Entrega de la versión final

Existe un bucle que va desde la fase c) hasta la fase f), en donde se desarrollan versiones, se hace entrega al cliente, el cual confirma si la versión es la deseada o por el contrario hay que realizar cambios. Este proceso conlleva mucho tiempo, pero por el contrario se consigue un producto totalmente personalizado y al máximo acorde con las necesidades del cliente

Para este proyecto se precisarían un diseñador y un Analista-Programador, pero la realidad es que todos los roles los voy a realizar yo, por lo que se reduce el número de personas a uno.

A continuación se muestra el gráfico Gantt donde se ven las relaciones entre las tareas llevadas a cabo y la distribución en el tiempo de las mismas, así como los recursos destinados a trabajar en las distintas fases del proyecto:

2.8.2.- Diagrama Gantt

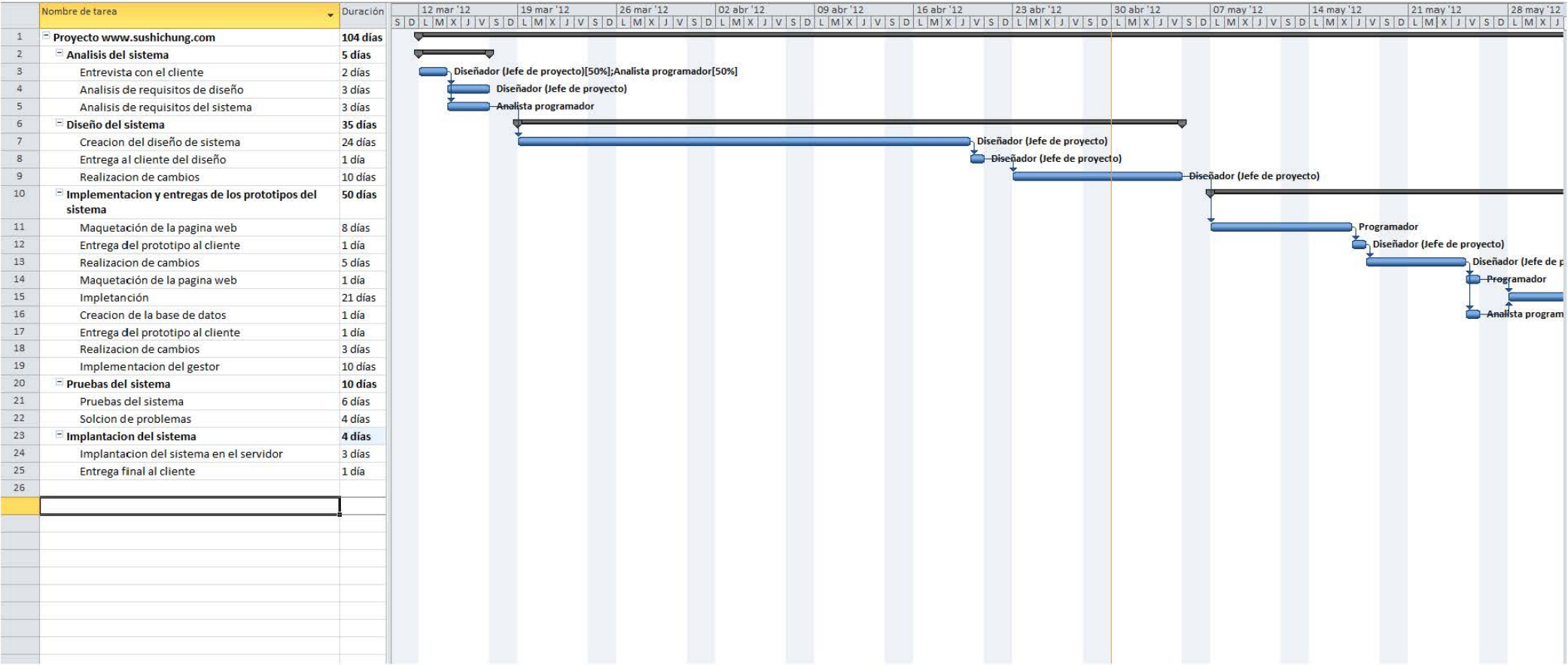


FIGURA 20 Diagrama Gantt parte 1

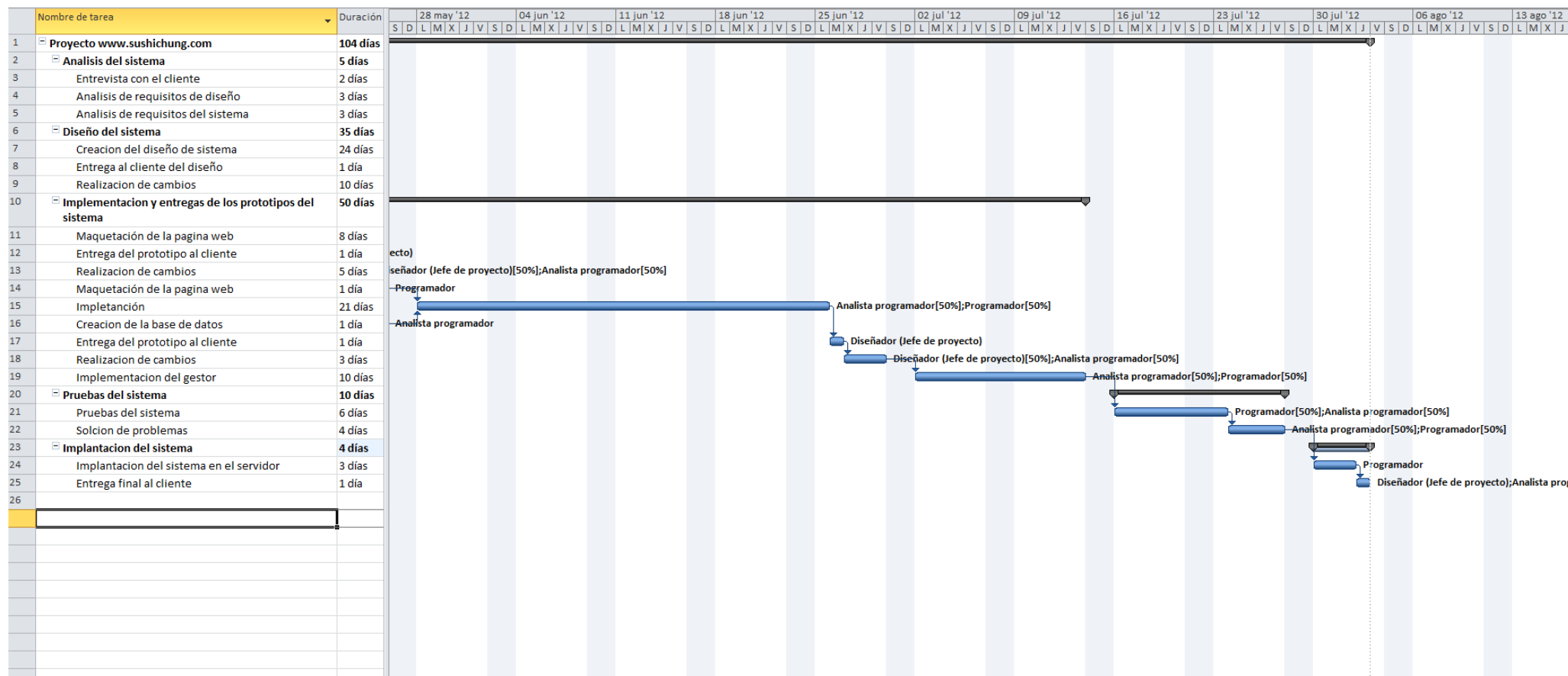


FIGURA 21 Diagrama Gantt parte 2



3.- DISEÑO

3.1.- Introducción

A través de los requisitos establecidos se contemplan las diferentes opciones.

Se contempla la idea de la navegación lateral como alternativa a la navegación vertical. Para comprobar la viabilidad de la idea se buscan paginas web que contengan este tipo de navegación, de esta manera podemos analizar los resultados obtenidos. Nos encontramos un estilo de web impactante, buenos diseños y una buena navegación.

El punto más importante de este análisis es saber de antemano si este tipo de páginas web tienen una buena navegación, que sea intuitiva para los usuarios.

El resultado del análisis es positivo y se establece la navegación de izquierda a derecha como requisito imprescindible, ya que otorga una visión nueva. Para facilitar la navegación por la página web y hacerla más intuitiva se concreta la necesidad de tener un menú que este siempre a la izquierda del explorador en donde se han de encontrar las diferentes secciones de la página web. De esta manera se puede llegar a cualquier parte de la web de manera rápida y sencilla.

Se precisa de un banner al comienzo de la web, en donde se detalla la última información relacionada con la franquicia o con alguno de los establecimientos.

A continuación están las diferentes partes de las que está compuesta la web, a través de los apartados, explicados anteriormente en los requisitos. Toda la información ha de ser útil, lo más breve posible, fácil de leer e interesante, para atraer la atención del lector.

En el diseño son cruciales los colores que se van a utilizar, que combinen, atraigan a los usuarios y sean agradables a la vista. Este es un punto crucial para garantizar el éxito de la web.

También es de suma importancia el logotipo de la franquicia, una vez elegidos los colores es hora de diseñar un logo que atraiga y sea fácil de ver y memorizar.

En cuanto a los apartados que componen la web, cada uno consta de un diseño particular. Con esto se consigue diferenciar cada sección de las otras. Por si esto no fuera suficiente para diferenciar las secciones de la web se introduce una pequeña película en flash entre los apartados. Ha de ser pequeña y que no atraiga al usuario, simplemente que se vea y sirva para diferenciar las secciones.

Algunos apartados cuentan con un pequeño menú. En el caso de tenerlo, este tiene un diseño en particular y su propia funcionalidad.

Una vez se genera el diseño de la web, este se entrega al cliente con el fin de evitar cualquier tipo de diseño o funcionalidad de la web que no sea la que el cliente quiere. Una vez aceptado el diseño se pasa a la parte de implementación.

3.2.- Disposición de las páginas

3.2.1.- Introducción

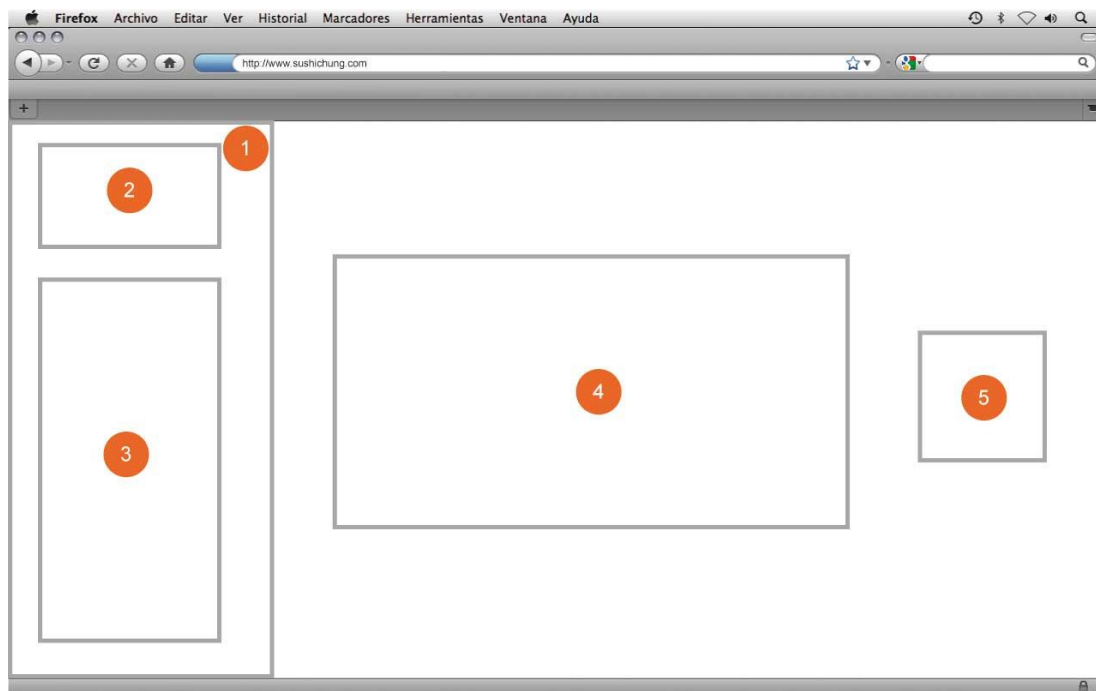
A continuación se muestra como van a ser distribuidos los diferentes componentes de los que esta compuesto el portal web comercial. Para hacerlo de una manera estructurada se van a mostrar los patrones de diseño de la página web principal y de sus secciones, seguidamente se pasara a mostrar el patrón de diseño del gestor de la misma manera que con la página principal.

Existirán siete tipos de patrones diferentes: inicio de la página principal (con el banner), sección estática de la página web, sección dinámica de la página web, pagina de inicio del gestor, pagina para la gestión de las secciones de la página web y la pagina para la creación y modificación de elementos de una sección.

A continuación se muestran las figuras que representan el patrón de diseño de cada tipo de página junto con los módulos que pertenecen al tipo de página correspondiente.

3.2.2.- Inicio de la página principal

Este tipo de disposición se utilizará únicamente para el inicio de la página web principal.



- 1 Menu de la página web. Este menu tiene una posición fija en el explorador independiente del contenido.
- 2 Enlace a la página web principal con el logotipo de la franquicia.
- 3 Enlaces a las secciones de la pagina web. Cada enlace tiene el nombre de la sección correspondiente.
- 4 Banner donde se mostrara la informacion más reciente de la franquicia.
- 5 Pelicula en Flash. Su función es separar visualmente las secciones entre si.

FIGURA 22 Diseño de tipo “Inicio de la página principal”.

3.2.3.- Sección estática

Este tipo de disposición se utilizará en las secciones estáticas de la página web.

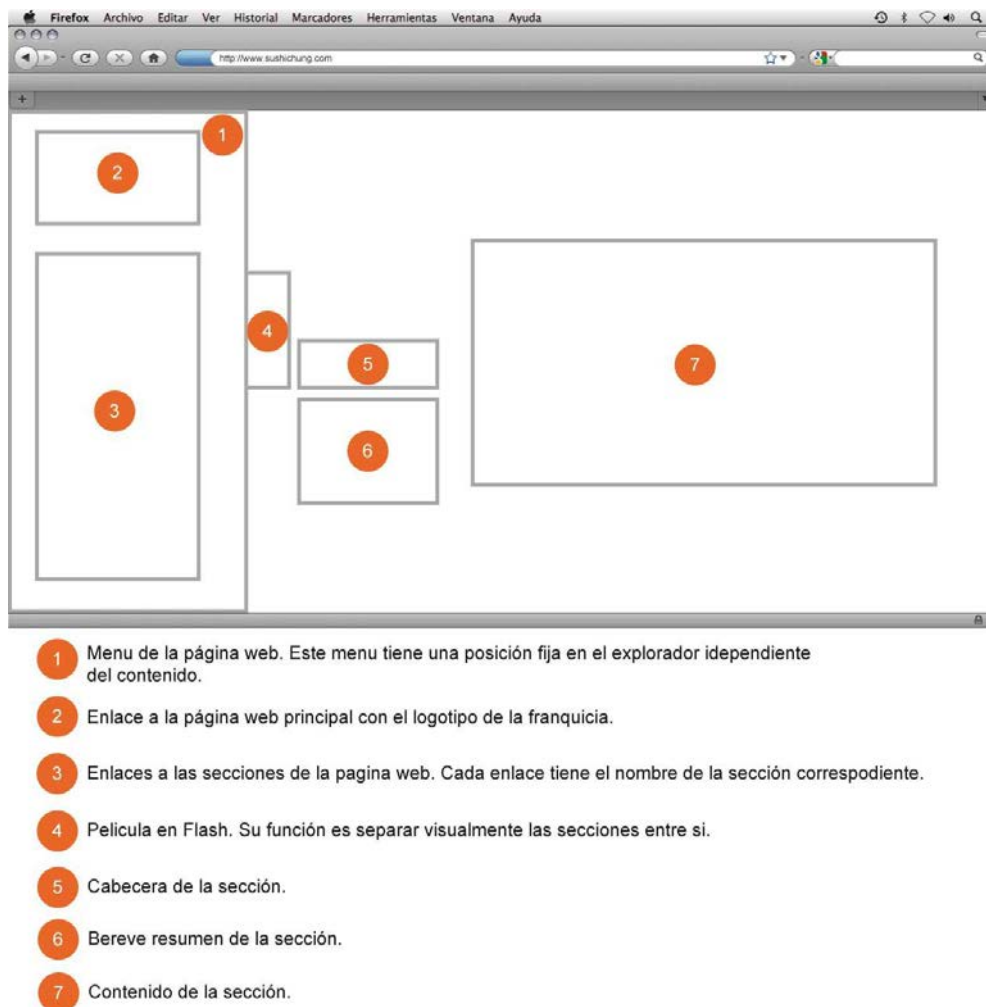
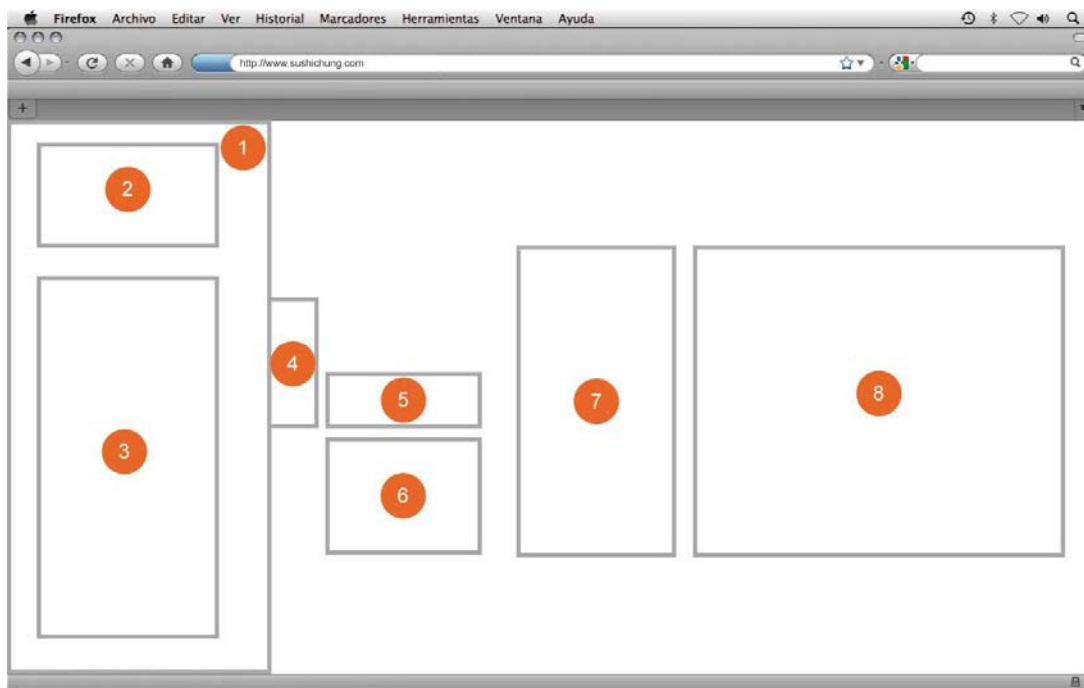


FIGURA 23 Diseño de tipo “Sección estática”.

3.2.4.- Sección dinámica

Este tipo de disposición se utilizará en las secciones dinámicas de la pagina web (Restaurantes y Noticias).

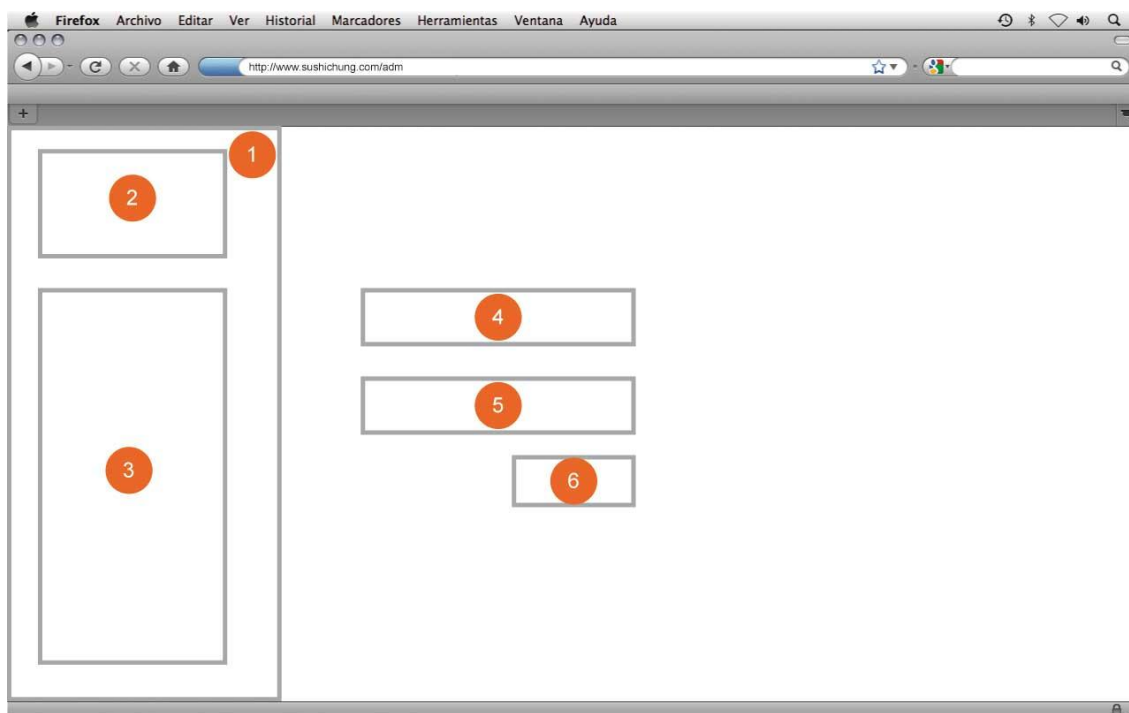


- 1 Menú de la página web. Este menú tiene una posición fija en el explorador independiente del contenido.
- 2 Enlace a la página web principal con el logotipo de la franquicia.
- 3 Enlaces a las secciones de la pagina web. Cada enlace tiene el nombre de la sección correspondiente.
- 4 Película en Flash. Su función es separar visualmente las secciones entre si.
- 5 Cabecera de la sección.
- 6 Breve resumen de la sección.
- 7 Índice del contenido de la sección.
- 8 Contenido de la sección.

FIGURA 24 Diseño de tipo “Sección dinámica”.

3.2.5.- Pagina de inicio del gestor

Este tipo de disposición solo se utilizará en la página principal de inicio del gestor.

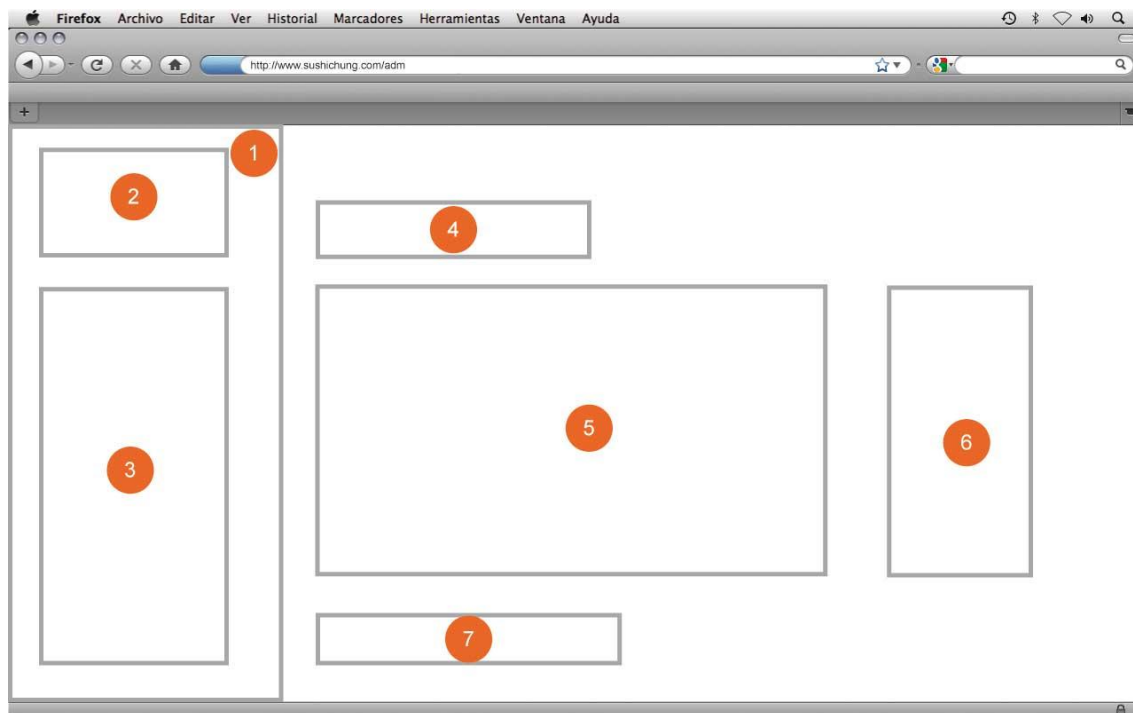


- 1 Menu del gestor.
- 2 Enlace a la página principal del gestor con el logotipo de la franquicia.
- 3 Enlaces a las secciones del gestor. Cada enlace tiene el nombre de la sección correspondiente.
- 4 E-mail del usuario para acceder al gestor.
- 5 Contraseña del usuario para acceder al gestor
- 6 Boton para logearse en el gestor.

FIGURA 25 Diseño de tipo “Inicio gestor”.

3.2.6.- Página de gestión de las secciones

Esta disposición se empleara en todas las paginas de gestión de las secciones gestionables.

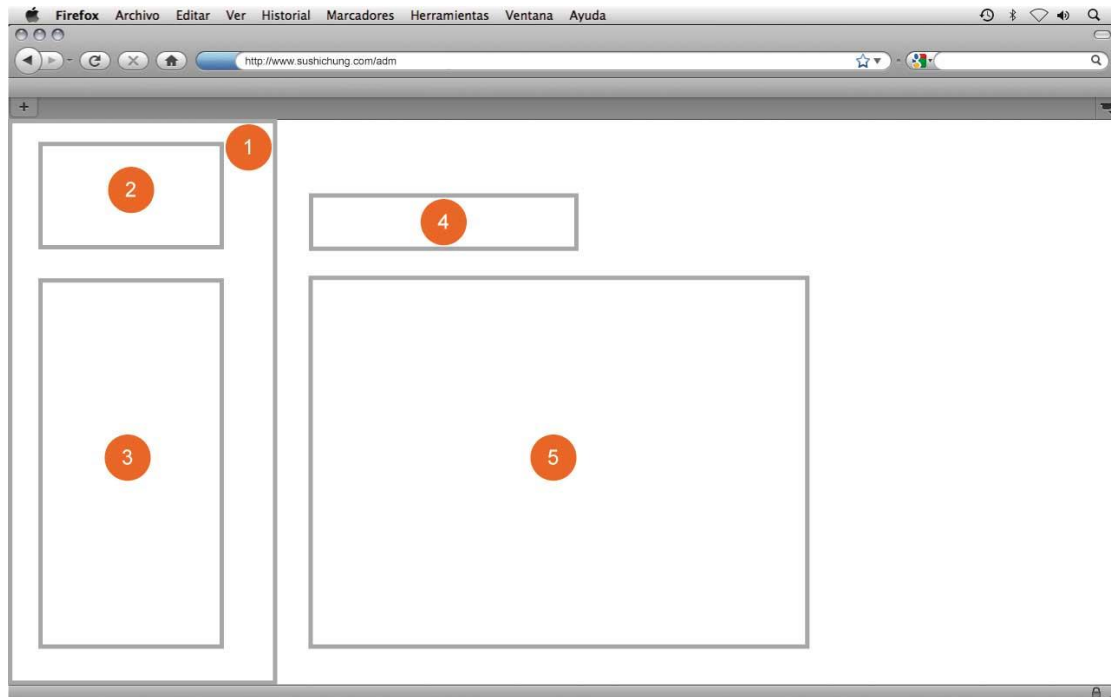


- 1 Menu del gestor.
- 2 Enlace a la página principal del gestor con el logotipo de la franquicia.
- 3 Enlaces a las secciones del gestor. Cada enlace tiene el nombre de la sección correspondiente.
- 4 Titular de la sección.
- 5 Índice con los elementos de la sección.
- 6 Enlaces editar y borrar para cada elemento de la sección.
- 7 Enlace para crear un nuevo elemento de la sección.

FIGURA 26 Diseño de tipo “Página de gestión de las secciones”.

3.2.7.- Página de creación y modificación de elementos de una sección

Esta disposición se va a utilizar en todas las páginas de creación y modificación de todas las secciones.



- 1 Menu del gestor.
- 2 Enlace a la página principal del gestor con el logotipo de la franquicia.
- 3 Enlaces a las secciones del gestor. Cada enlace tiene el nombre de la sección correspondiente.
- 4 Titular de la sección.
- 5 Formulario con los datos de un elemento de la sección.

FIGURA 27 Diseño de tipo “Página de creación y modificación”.

3.3.- Diseño de la base de datos

3.3.1- Introducción

En este apartado de la memoria se va a detallar el diseño de la base de datos. Para ello se crea el diagrama de entidad/relación en donde se representan las entidades relevantes del portal web comercial, así como sus interrelaciones y propiedades en el modelo relacional, donde todos los datos son almacenados en relaciones y cada relación es un conjunto de datos. También se muestra el diagrama de clases, representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

3.3.2- Diagrama Entidad/Relación

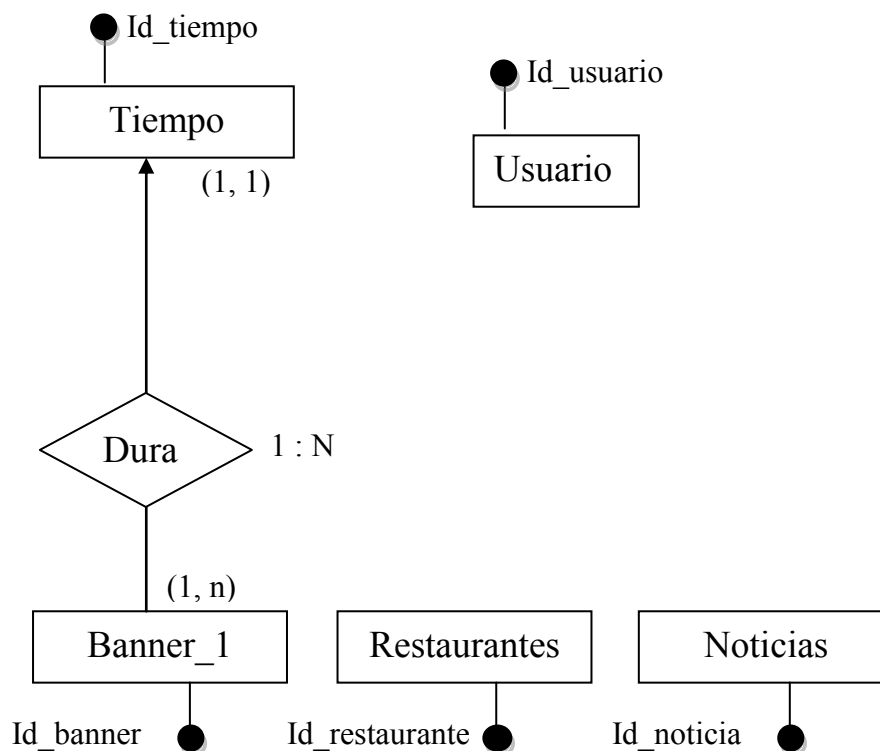


FIGURA 28 Diagrama entidad/relación

3.3.3.- Determinación de las entidades

Es una parte del diseño de la base de datos muy importante y requiere de un buen análisis de los requisitos, ya que será determinante para la realización de una buena base de datos consistente y adecuada para cubrir los objetivos del proyecto. Se van a definir las entidades conceptuales de la base de datos del sistema y las relaciones existentes entre dichas entidades conceptuales.

En caso de realizar un diagrama de Entidad/Interrelación incorrecto, la base de datos por consecuencia también lo sería, la razón es que a partir del diagrama de Entidad/Interrelación se obtiene el grafo relacional que será el implementado para la base de datos.

Si observamos el diagrama de Entidad/Interrelación vemos que solo existe una única relación en todo el diagrama de Entidad/Interrelación. Se debe a que la relación entre usuarios y secciones de la pagina web (banner, restaurantes y noticias) se lleva a cabo en el gestor, en donde los usuarios pueden crear, modificar o eliminar contenido de las secciones de la pagina web.

A continuación se especifica cada una de las entidades:

- Usuario: es el usuario gestor, determina cual será el contenido de las secciones de la página web.
- Banner: referencia al contenido del banner principal de la página web.
- Restaurantes: determina la cantidad y la información de los restaurantes registrados en el sistema, los cuales serán mostrados en la sección restaurantes de la página web.
- Noticias: contiene todas las noticias almacenadas en el sistema, las cuales están relacionadas con la franquicia y serán mostradas en la sección noticias de la página web.
- Tiempo: indica el tiempo transcurrido en el paso de diapositivas de cada banner.

3.3.4- Determinación de las relaciones existentes entre las entidades

Dura: esta relación se creo como previsión al futuro. La idea es que pueden aparecer varios banners en la página web. En cada uno de ellos el tiempo transcurrido del paso de una diapositiva a otra puede ser diferente y hace falta una tabla que contenga el tiempo de cada uno, y una tabla para cada banner.

3.3.5.- Supuestos semánticos no reflejados

Un usuario (gestor) registrado en la base de datos nunca podrá acceder a la tabla usuarios, esta tabla es gestionada por los administradores a través del acceso directo a la base de datos.

3.3.6.- Modelo Relacional

Usuarios (id, email, pass)

Banner_1 (id, contenido, posición_texto, fondo, posición_banner, id_tiempos)

Restaurantes (id, nombre, entrada_indice, titulo, dir_foto, teléfono, dirección, horario_mañana, horario_tarde, horarios_centro)

Noticias (id, fecha, entrada_indice, foto, titulo, fuente, contenido, link, activo)

Tiempos (id, nombre, tiempo)

FIGURA 29 Modelo relacional

3.3.7.- Diagrama de clases

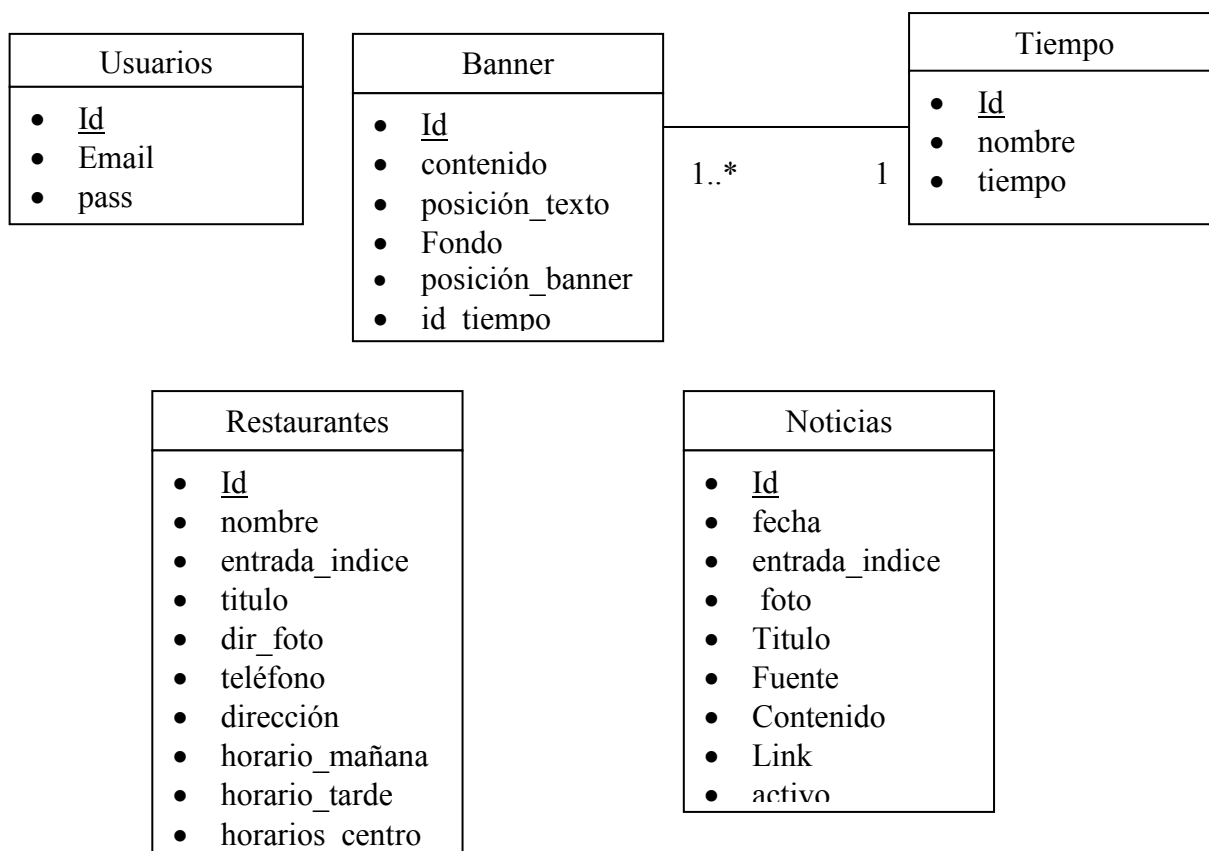


FIGURA 30 Diagrama de clases



4.- DESARROLLO

4.1. Introducción

La implantación se realiza mediante programación por capas, siguiendo una arquitectura de tres capas. Es un estilo de programación cuyo objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos.

La ventaja principal se encuentra en el desarrollo, se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, facilita la adaptabilidad y extensibilidad del proyecto.

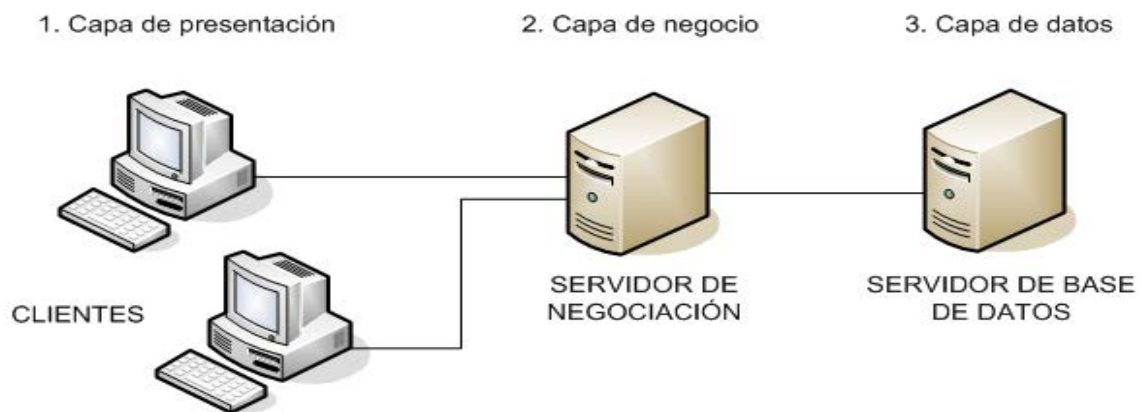


FIGURA 31 Definición gráfica de la programación por capas

- **1) Capa de presentación:** Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" para el usuario, generalmente se presentan como formularios (realiza un filtrado previo para comprobar que no hay errores de formato).
- **2) Capa de negocio:** Es la capa que recibe las peticiones del usuario y envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para recuperar, crear, modificar o borrar información de la base de datos.

- **3) Capa de datos:** Es donde residen los datos y es la encargada de acceder a los mismos. Formada por la base de datos que realiza todo la creación, el almacenamiento, modificación y eliminación de datos. Reciben las solicitudes desde la capa de negocio.

En la capa de presentación esta el navegador que permite visualizar la página web. Él mismo que se comunica con el servidor Web y a su vez con el servidor de aplicaciones conformando la lógica de negocios y posteriormente se accede a la Base de datos.

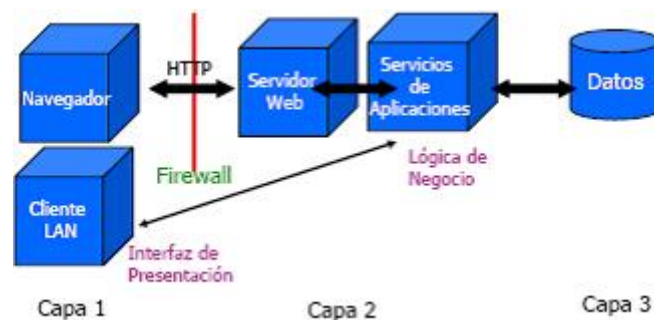


FIGURA 32 Esquema de la programación por capas

El servidor de la aplicación actúa como enlace entre los servidores Web y el servidor de la Base de datos. Se consigue fomentar la escalabilidad del sistema, la seguridad y la reutilización de código.

4.2.- Mapa web

4.2.1.- Introducción

El mapa web del portal web comercial desarrollado en este proyecto de fin de carrera muestra una lista de las páginas web que componen la aplicación web (pagina web principal y pagina web para el gestor de contenido) y las secciones que las componen, organizadas de forma jerárquica:

4.2.2.- Mapa web

Mapa web de la página principal.

- Pagina web
 - Smart food
 - Siempre fresco!
 - Sushi
 - Restaurantes
 - Noticias
 - Contacto
 - Franquicia

Mapa web del gestor de contenido:

- Login
 - Gestor de restaurantes
 - Crear restaurante
 - Editar restaurante
 - Gestor noticias
 - Crear noticia
 - Editar noticia
 - Gestor del banner
 - Crear diapositiva
 - Editar diapositiva

4.3.- Capa de presentación

4.3.1.- Introducción

Para explicar la capa de presentación se muestra primero la página principal de la página web, accesible por todos los usuarios que lo deseen. La página principal esta compuesta por varias secciones, por lo tanto se realizara una explicación detallada de cada sección, para que es, como funciona y como se comunican con la capa de negocio.

Una vez terminada la explicación de la página principal se procederá a realizar el mismo proceso para el gestor, empleado el mismo método, se irán presentando una a una cada una de las secciones de las que esta compuesto el gestor, explicando cada sección, para que es, como funciona y como se comunican con la capa de negocio.

4.3.2- Página principal

Antes de mostrar las diferentes secciones de la página principal voy a explicar detalles de la página principal.

4.3.2.1.- Menú

Lo primero que encontramos al entrar en la página web principal es un menú en donde se muestran todas las secciones de la página web.



FIGURA 33 Menú web

El nombre de cada sección es un enlace que ejecuta dos funciones de jQuery:

- `scrollTo()`: se invoca a través de la librería de jQuery `jquery.scrollTo-1.4.2.js`. La función se ejecuta en los enlaces del menú, cada enlace desplaza el contenido de la página web hasta llegar a la sección indicada en el enlace, es decir, esta función realiza un `scroll` de la página web en el eje x hasta llegar a la sección indicada en el enlace del menú.
- `removeClass ()`: esta función de jQuery cambia la clase de la entidad deseada. Su función es cambiar la clase, en este caso cambia la clase de los enlaces del menú. Estos enlaces tienen dos posibles clases, la clase “`menu_general_activo`” o ninguna clase. En caso de que algún enlace tenga la clase “`menu_general_activo`” el color de las letras pasará a ser magenta y aparecerá una pequeña flecha magenta a la derecha del enlace, indicando al usuario la última sección seleccionada, que suele ser la sección en donde se encuentra el usuario.

Esta misma función se aplica al pasar el ratón por encima del enlace, en cuyo caso cambia a la clase “`menu_general_activo`”, a no ser que el enlace posea ya esta clase. Cuando el ratón deja de estar encima del enlace se elimina la clase, a menos que el enlace esté activo.

4.3.2.2.- Sección Banner

Esta sección consta de un banner formado por varias diapositivas, en donde cada diapositiva es cargada desde la base de datos. Para poder cargar la información desde la base de datos la capa de negocio recoge la información de la capa de datos. Las transiciones entre la capa de negocio y la capa de datos se verán mas adelante. Basta con saber que la información de las diapositivas del banner están almacenadas en la base de datos y son gestionables desde el gestor.

Para captar rápidamente la atención del usuario el banner posee una posición privilegiada en la página web. Se encuentra al comienzo de la misma y su función es proporcionar al usuario información que le resulte interesante sin que el usuario tenga que buscar dicha información por la página web. La información de cada diapositiva del banner es manejable por el usuario gestor a través del gestor.

El banner funciona mediante la librería “jquery.vslider.js”. Esta librería crea de manera sencilla un banner con tantas diapositivas como se desee, basta con crear un menú en html, en donde cada apartado del menú corresponderá a una diapositiva en el banner, y ejecutar una simple función en JavaScript correspondiente a la librería jquery.vslider.js. Posee varios efectos para el paso de una diapositiva a otra. Para este banner se ha empleado un único efecto.



FIGURA 34 Diapositiva del Banner con el texto a la izquierda



FIGURA 35 Diapositiva del Banner con el texto a la derecha

En la imagen se muestra una de las diapositivas creadas para probar el banner. Se pueden observar a cada lado de la diapositiva del banner unos pequeños botones con una flecha en su interior. Estos botones se usan para pasar de diapositiva en el banner de manera manual. Estos botones solo parecen cuando el ratón pasa por encima del banner, de esta manera se evita la pérdida de información por superposición de los botones sobre el texto o la imagen de cualquiera de las diapositivas.

Ambas imágenes poseen texto y una imagen. La posición del texto y de la imagen se determina en el gestor. Se ofrecen dos posibilidades para el posicionamiento del texto, a la derecha o la izquierda de la diapositiva del banner. La imagen de fondo es elegida por el usuario gestor, y debe ser acorde con la posición indicada para el texto elegido.

4.3.2.3.- Sección Smart Food

Esta sección, como las secciones restantes, consta de una pequeña película en flash al comienzo de la sección. Esta película se emplea para diferenciar una sección de otra. Después de la pequeña película en flash se encuentra la cabecera de la sección y seguidamente la sección en sí.

Esta sección tiene una función puramente informativa de la franquicia. En donde se detalla información sobre la franquicia exclusivamente.

La información de esta sección es estática y por lo tanto el contenido de la sección se encuentra en el código html de la página web, por lo que no precisa de acceso a la base de datos para cagar información.



Sushi Chung es un nuevo concepto en alimentación, que engloba **salud**, **lifestyle** y una visión vanguardista de empresa.

Salud, por ser una dieta equilibrada, completa, cosmopolita, que recoge los secretos de la cocina tradicional japonesa, y los adapta a las necesidades del consumidor moderno que necesita algo más que alimentarse... Vitaminas, Omega-3, hidratos de carbono, yodo y proteínas se combinan en un concepto de alimentación fresco y saludable.

Lifestyle en cada expresión de nuestra empresa. Nuestro sushi es calidad, servido con lo mejor del concepto fast food... Inmediatez, un menú completo, a la carta, y con precios amigables. Ofrecemos una experiencia completa de alimentación. Una opción inteligente para nuestros clientes...

La filosofía **Smart food** alcanza todos los ámbitos de nuestra empresa. **Sencilla, funcional, pero a la vez sofisticada e inteligente.**

Así son nuestros **Restaurantes Sushi Chung**, en los que podrás encontrar un lugar agradable para consumir nuestros productos, adquirirlos en cómodos menús "take away", o encargarnos a través de nuestro servicio "delivery". Descubre nuestro ingenioso packaging easy-open, que permite una óptima conservación del producto en porciones individuales, preservando la calidad y frescura tan propia de nuestro sushi.

"Ofrecemos una experiencia completa de alimentación. Una opción inteligente para nuestros clientes"

FIGURA 36 Sección SmartFood

4.3.2.4.- Sección Siempre fresco!

Se repite la estructura de comienzo de la sección, primero una pequeña película en flash y a continuación la cabecera de la sección.

Esta sección esta formada por una imagen, en la que se detalla el proceso de elaboración del sushi en la franquicia pasó a paso, y de esta manera eliminar al usuario posibles dudas respecto al tema.

La información de esta sección es estática y por lo tanto el contenido de la sección se encuentra en el código html de la página web, por lo que no precisa de acceso a la base de datos para cagar información.



FIGURA 37 Sección Siempre fresco!

4.3.2.5.- Sección Sushi

El comienzo de esta sección esta compuesta por una película en flash, una cabecera y el contenido. En el contenido de la sección se explican los diferentes sushis existentes, los ingredientes que contiene y la manera en la que se prepara. La función de esta sección es informar a los usuarios menos aventajados en el conocimiento del sushi. La carta esta compuesta por más platos, pero se detalla la información acerca del sushi por ser el plato principal de la franquicia y por el frecuente desconocimiento por parte de los usuarios en la materia.

La información de esta sección es estática y por lo tanto el contenido de la sección se encuentra en el código html de la página web, por lo que no precisa de acceso a la base de datos para cagar información.



FIGURA 38 Sección Sushi

4.3.2.6.- Restaurantes

Como viene siendo habitual, en la parte izquierda de esta sección se encuentra una película en flash y la cabecera de la sección. A continuación esta el contenido de la sección, en donde se muestran los restaurantes de la franquicia ya abiertos y las próximas aperturas de restaurantes.

Esta sección requiere del acceso a la capa de datos, a través de la capa de negocio, para recuperar la información de los restaurantes, tanto si son restaurantes abiertos como si no.

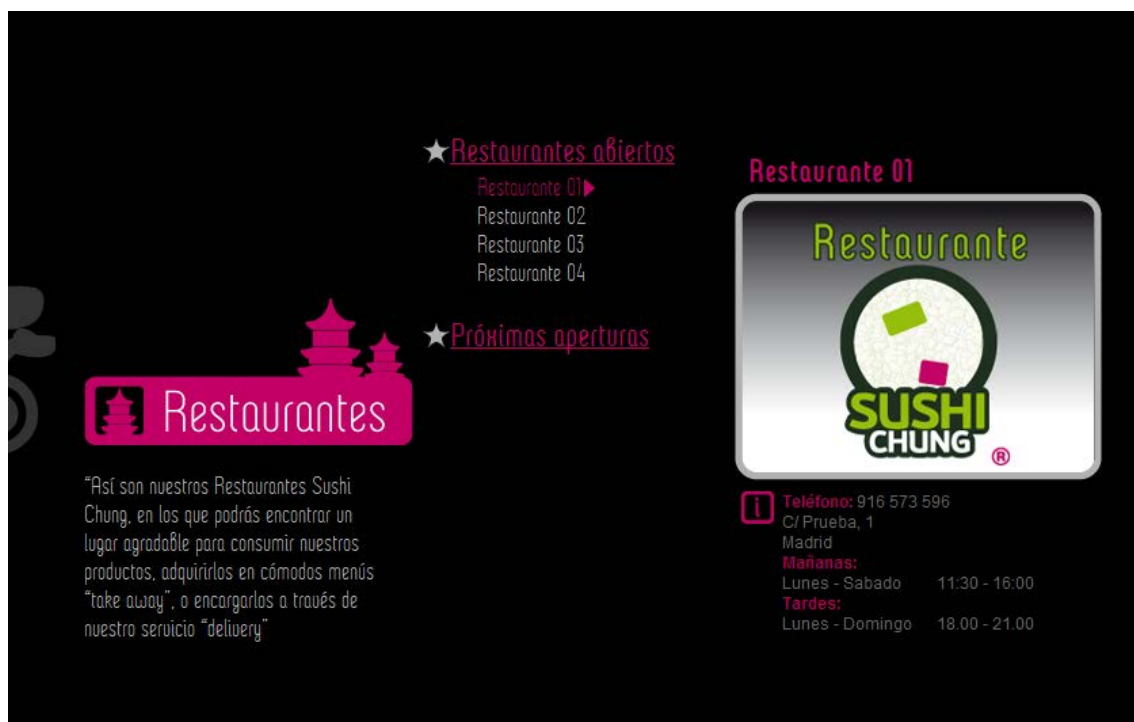


FIGURA 39 Menú restaurantes abiertos

La sección esta compuesta por dos índice, un índice para los restaurantes abiertos (primera imagen) y otro para las próximas aperturas (imagen de debajo). Solo se permite mostrar uno de los índices, nunca los dos a la vez. El motivo es el espacio, con este sistema se permite mostrar mas contenido de cada índice que si se mostraran los dos índices juntos. Sin embargo siempre están visibles los nombre de los dos índices para poder escoger uno ellos, basta con pinchar en uno de los titulares de los índices que se muestran para desplegar el índice con los restaurantes que alberga. Por defecto esta abierto el índice correspondiente a los restaurantes abiertos por ser de mayor interés.



FIGURA 40 Menú restaurantes próximas aperturas

Dentro de los índices aparece el nombre del lugar donde se encuentran los restaurantes o en donde se va a abrir próximamente uno. Para ver la información de los restaurantes abiertos basta con seleccionarlo en el índice que corresponda.

La información de cada restaurante se muestra a la derecha de los índices. Contiene un título para el restaurante, una foto, teléfono, dirección y horarios de apertura por la mañana y por la tarde.

En algunos casos aparece un pequeño texto legal justo al final de la información del restaurante, esto es debido a que algunos restaurantes se encuentran en centros comerciales, y por lo tanto el restaurante que allí se encuentra está sujeto a los horarios y días laborables del centro comercial.

4.3.2.7.- Noticias

La sección noticias contiene una película en flas, una cabecera, dos índices y una noticia visible.

En esta sección se muestran las noticias relacionadas con franquicias, con el sushi y con la propia franquicia para la que se desarrolla el portal web.

Existen dos índices, el primero son las noticias publicadas en el mes actual, y el otro índice contiene el resto de las noticias pasadas. Se muestra siempre una noticia por defecto, siendo esta la noticia destacada.



Este mes en Sushi Chunger...

- Noticia Prueba 01
- Noticia prueba 02

En meses anteriores...

- El córner, la solución a la falta de espacio de las franquicias
- Sushi, una dieta equilibrada ▶

Noticias

"Te ponemos al día de lo que acontece en Sushi Chunger"

Sushi, una dieta equilibrada

Fuente: <http://www.directoalpaladar.com>
> 07/Septiembre/2011

La dieta japonesa, según los estudios publicados, siempre aparece como una de las más sanas, con unos índices de mortalidad cardiovascular muy inferiores al resto de los países, sobre todo si lo comparamos con países occidentales del centro y norte de Europa, o EEUU. Una de las causas demostradas de esta baja mortalidad es sin duda el alto consumo de pescado, arroz y verduras de este país, muy por encima incluso de los países mediterráneos, unido a un bajo consumo de grasas de todo tipo y de carnes. El sushi es el prototipo de esta dieta sana. Sólo viendo sus ingredientes básicos podemos comprenderlo. Arroz hervido, pescado, verduras, soja y algas marinas. Sano, ligero y además un aporte equilibrado de proteínas, hidratos de carbono, fibra, vitaminas y minerales. Libre de grasas animales y rico en omega-

[Leer más...](#)

FIGURA 41 Noticia con foto

Toda noticia consta de un titular, la fuente de la noticia, pudiendo ser la propia franquicia, la fecha de publicación de la noticia, una foto, el contenido de la noticia y un link a la fuente de la noticia. Este último link se muestra un enlace en el que se puede leer "Leer más". Si el usuario pincha en este enlace se abrirá una nueva pestaña en su explorador con la dirección de la fuente de la noticia.

Las noticias tiene la opción de mostrar una foto, pudiendo también no mostrarla. Lo mismo ocurre con el link ala fuente de la noticia, puede aparecer o no. La razón es sencilla, algunas noticias carecen de foto, y en el caso del link es posible que la notica sea propia de la franquicia y por lo tanto no requiere de link externo.



FIGURA 42 Noticia sin foto

4.3.2.8.- Contacto

Mismo comienzo de sección, una pequeña película en la flash a modo de separador de secciones y una cabecera con el título de la sección un pequeña introducción debajo.



FIGURA 43 Sección contacto

En esta sección los usuarios pueden enviar consultas y sugerencias al correo de la franquicia. La idea es que los usuarios puedan realizar cualquier consulta acerca de la franquicia, algún restaurante, la carta... También pueden enviar sugerencias, ayudando a mejorar la franquicia.

La sección está formada por un formulario, antes de enviarlo se comprueba que todos los campos del formulario no estén vacíos y se marca la casilla “He leído y acepto las condiciones de uso”, y en el caso del campo e-mail se comprueba que el e-mail tenga un formato de correo. Si todos los campos son correctos se pasa el formulario a la capa de negocio, la cual se verá más adelante como envía el formulario a un correo.

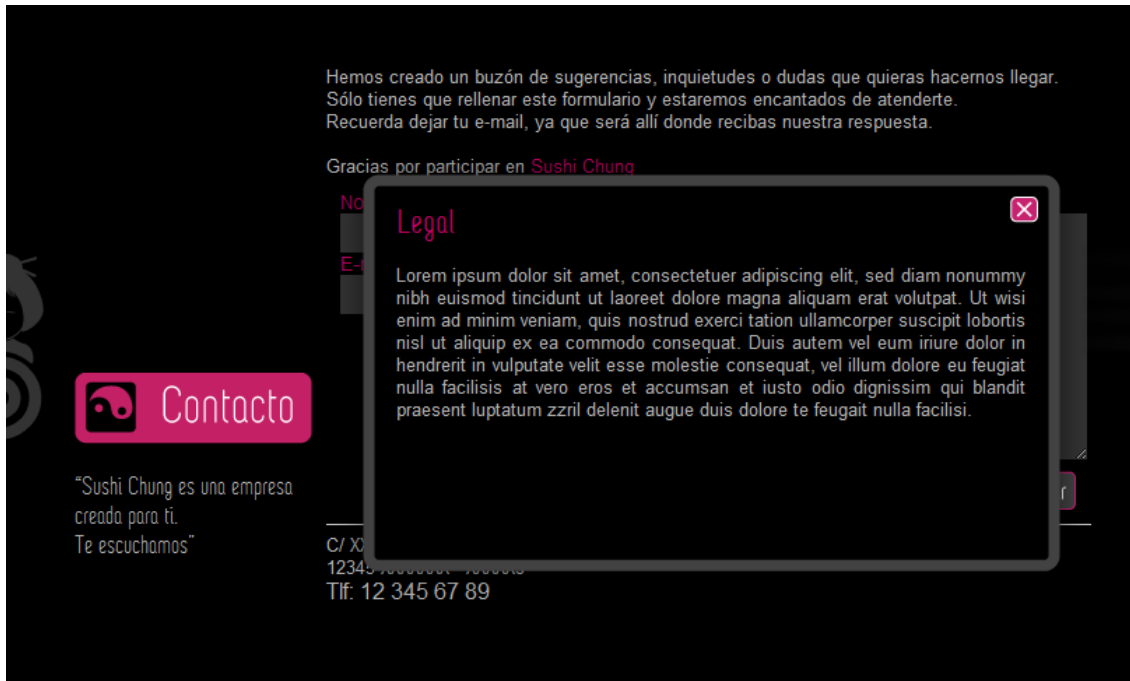


FIGURA 44 Aviso legal

En la casilla “He leído y acepto las condiciones de uso” existe enlace en las palabras “ condiciones de uso” de color magenta. En caso de que el usuario quiera ver las condiciones legales que debe aceptar para enviar una consulta debe pinchar en este enlace. Cuando se pulsa el enlace, este llama a una función de jQuery, la cual hace visible un cuadrado con fondo negro y un recuadro gris que contiene el texto legal. La función de jQuery además de hacer visible el cuadrado lo hace visible con efecto “persiana”, es decir no aparece de repente, si no que aparece de arriba hacia abajo con rebote al llegar abajo.

El cuadrado posee un fondo, en caso de que el usuario pinche en el fondo, el cuadro desaparece de la misma manera en la que apareció, pero a la inversa, es decir se oculta de debajo hacia arriba. Puede parecer que el hecho de poner en el fondo del recuadro un enlace para cerrar el cuadrado del texto legal no sea muy intuitivo, por ello el fondo consta de un aspa a modo de botón de cierre de ventana en Windows, de esta manera el usuario interpretara que al pinchar en el aspa se cerrara el cuadrado con el texto legal.

4.3.2.9.- Franquicia

Mismo comienzo para esta sección que en las anteriores, una pequeña película en flash y una cabecera con el título de la sección y una pequeña introducción debajo.

Esta sección es para aquellos usuarios interesados en abrir un restaurante de la franquicia. Se marcan algunos detalles mínimos para la creación de un restaurante.

La sección cuenta con un botón cuya función para el usuario es la de descarga el dossier con toda la información relevante a la apertura de un restaurante de la franquicia. También se indica el email de contacto para posible franquiciados, el cual es un enlace con la función de JavaScript “mailto:” que contiene la dirección del correo destinado a recibir las solicitudes de los posibles futuros franquiciados.



Business to Be **more**

Franquicia

"Sushi Chung en España. Un nuevo concepto en alimentación, que engloba salud, lifestyle y una visión vanguardista de empresa."

Descárgate nuestro dossier y descubre cómo unirte a un proyecto de garantías.

[Descarga Dossier](#)

Departamento de Expansión:
123 456 789
info@Sushi_Chung.com

Llave en **Mano**
€ **Mínima Inversión**
Alta Rentabilidad

FIGURA 45 Sección Franquicia

4.3.3.- Gestor

El gestor es la parte del sistema creada para que los usuarios gestores gestionen el contenido de las secciones banner, restaurantes y noticias.

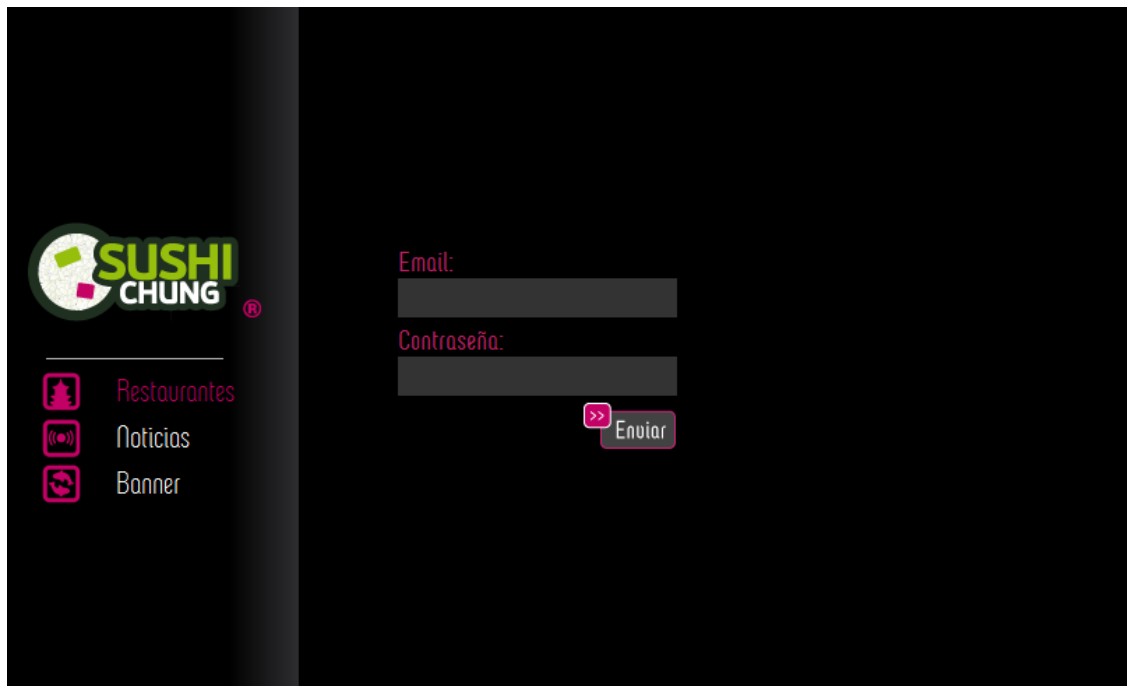


FIGURA 46 Página de inicio del gestor

Para acceder al gestor basta con introducir en un explorador de internet la dirección de la página web seguido de “/adm”. Por motivos de seguridad se obliga a los usuarios a ingresar su dirección de correo electrónico, con la que fue dado de alta, y su contraseña. Si es correcto se pasa a la primera sección del gestor. En caso de introducir datos que no se correspondan con la cuenta de un usuario gestor dado de alta en el sistema, se indica al usuario que el nombre y/o contraseña son incorrectos.

La seguridad del gestor a de ser robusta e impedir que accedan intrusos. El motivo es claro, el gestor es el medio a través del cual se crea, modifica o elimina contenido, en caso de una intrusión el contenido de las secciones gestionables corre peligro. Para mejorar la seguridad de la página los administradores de la página realizan cada cierto tiempo un backup de la base de datos.

A la izquierda aparece un menú muy similar al de la página web principal, pero solo con las secciones gestionables. En este caso el menú funciona como un menú normal, cada enlace del índice es un enlace a una página del gestor referente a la sección seleccionada. Además cuenta con un enlace para cerrar sesión, el cual redirige al usuario a la página principal del gestor.

Todas las secciones del gestor cuentan con 3 páginas, la página del gestor propio de la sección, en donde se encuentran todos los registros de la sección, la página de creación de un nuevo registro para la sección y la pantalla de modificación del registro seleccionada.

La página del gestor de cada sección contiene todos los registros de la tabla que le corresponde de la base de datos en una lista con todos los titulares o nombre de las entradas. A la derecha de cada titular se encuentran dos enlaces, el primero es enlace “Editar”, que como su propio nombre indica se emplea para editar un registro de la sección, para ello en enlace editar abre la página de modificación del contenido de un registro de la sección con todos los datos del registro ordenados en un formulario. A la derecha del enlace “Editar” está el enlace “Borrar”, el cual tiene la función de borrar el registro de la sección. En caso de querer borrar un registro se debe pulsar el enlace “Borrar” correspondiente al registro, a continuación se muestra un mensaje de confirmación para asegurar que se quiere borrar el registro, ya que el usuario puede pulsar este enlace por equivocación.

Algunas páginas del gestor contienen en sus formularios el campo contenido, para los cuales se han creado editores de texto en donde se pueden modificar muchas de las propiedades del texto que no pueden ser modificadas en un campo de texto en un formulario html. El editor de texto empleado para cada uno de ellos es el editor “Nice-Editors”:

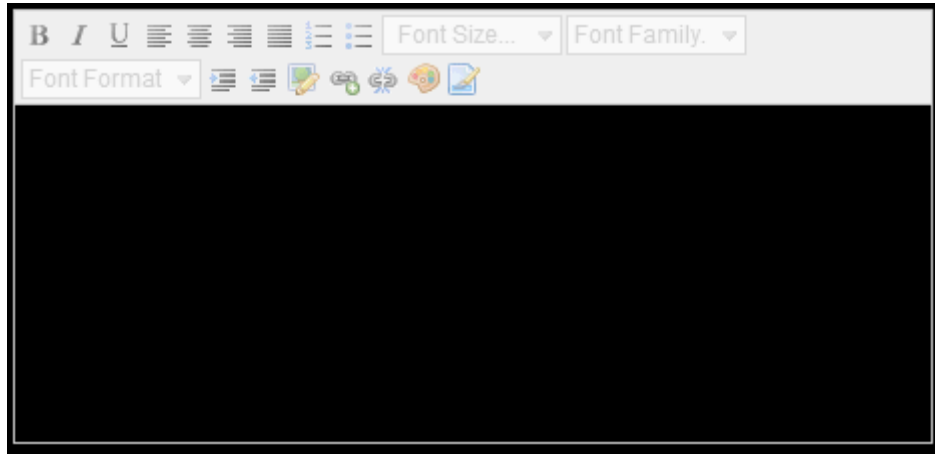


FIGURA 47 Editor de texto Nice Edit

Para crear este editor de texto es necesaria una librería de JavaScript (nicEdit.js). El editor se invoca mediante una sentencia de JavaScript, en donde se indica el id del campo del formulario en el que se desea implementar el editor de texto.

Permite manejar muchas de las propiedades del texto:

- Letra en negrita.
- Letra en cursiva
- Letra subrayada
- Posición del texto.
- Crear listas de viñetas
- Crear listas numeradas
- Cambiar el tamaño de la letra
- Cambiar la fuente de la letra
- El formato de la letra.
- Tabular líneas
- Introducir en el texto una imagen
- Introducir un enlace.
- Cambiar el color de la letra
- Cambiar el color de fondo de la letra

Alguno de los campos de los formularios de las páginas del gestor corresponden a fechas. Para mejorar este campo se ha implementado un botón al lado del campo fecha, cuando es pulsado se muestra un calendario en donde el usuario tiene que marcar la fecha deseada, una vez marcada en el campo del formulario correspondiente a la fecha se rellena. Si el usuario intenta introducir manualmente la fecha se muestra un mensaje de error en el que se le indica que debe presionar el botón del calendario para introducir la fecha.



FIGURA 48 Calendario

La razón por la que no se permite introducir la fecha al usuario se debe al formato de fechas empleado por SQL. El formato de fechas de SQL es un formato con restricciones, y por lo tanto no se debe permitir al usuario introducir la fecha manualmente para evitar fallos en la base de datos. Además el calendario facilita al usuario a introducir la fecha.

La fecha que aparece marcada cuando un usuario abre el calendario será siempre la fecha actual.

4.3.3.1- Gestor Restaurantes

La sección cuenta con dos tablas en donde se encuentran todos los restaurantes de la sección. La primera tabla referencia a los restaurantes abiertos, mientras que la segunda tabla corresponde a los futuros restaurantes que aun no han sido abiertos.

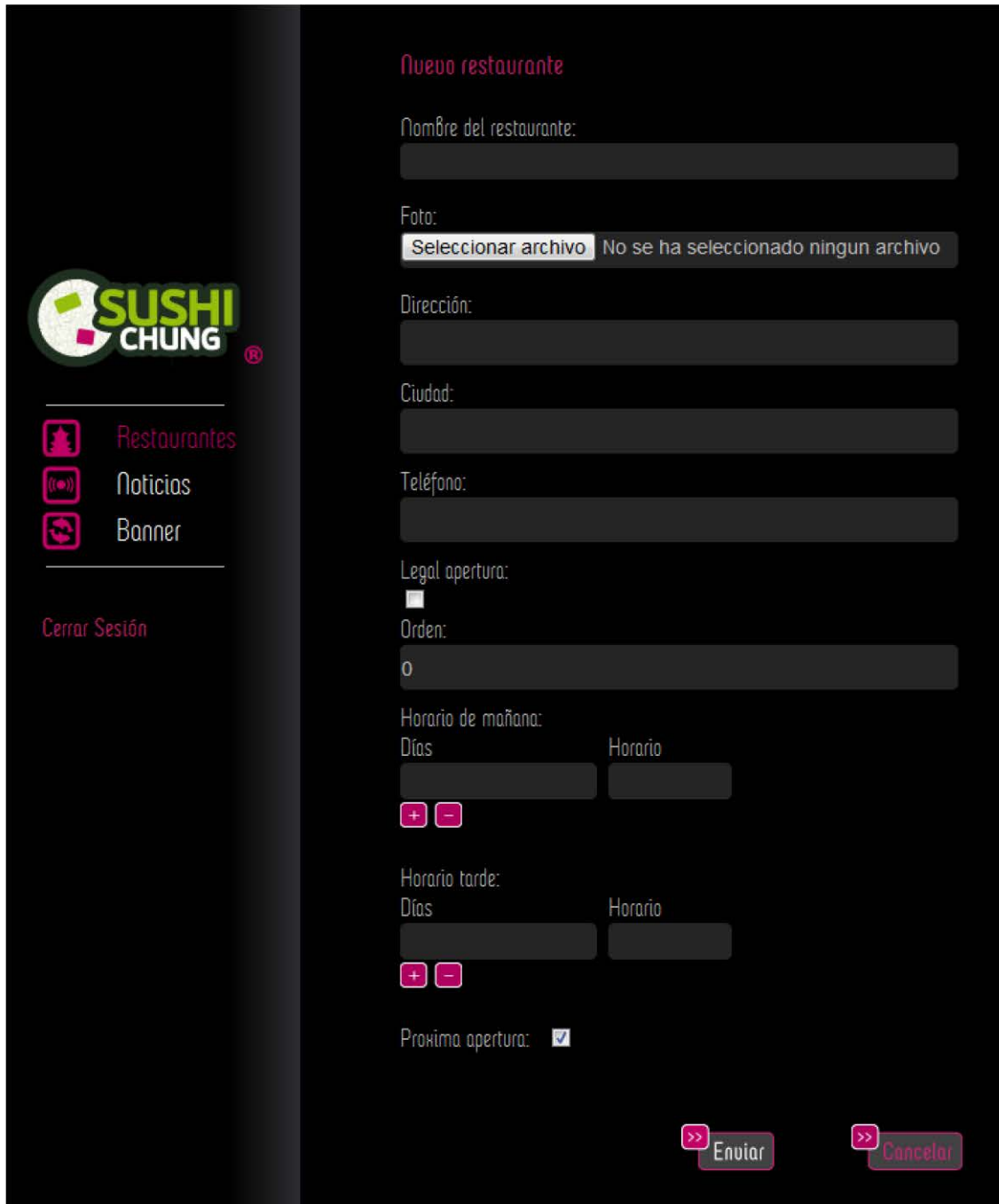
Debajo de las tablas se encuentra el enlace ">>Nuevo restaurante<<" el cual redirige a la pagina de creación de un nuevo restaurante en el sistema.



FIGURA 49 Gestor de restaurantes

4.3.3.2- Nuevo Restaurante

En esta sección se crean los nuevos restaurantes, tanto abiertos, como próximas aperturas.



The screenshot shows a web interface for creating a new restaurant. On the left is a dark sidebar with the 'SUSHI CHUNG' logo and a menu with three items: 'Restaurantes' (with a fork and knife icon), 'Noticias' (with a speech bubble icon), and 'Banner' (with a banner icon). Below the menu is a 'Cerrar Sesión' link. The main area is white and titled 'Nuevo restaurante' in red. It contains several form fields: 'Nombre del restaurante:' (text input), 'Foto:' (file upload button labeled 'Seleccionar archivo' with feedback text 'No se ha seleccionado ningun archivo'), 'Dirección:' (text input), 'Ciudad:' (text input), 'Teléfono:' (text input), 'Legal apertura:' (checkbox), 'Orden:' (text input with '0' entered), 'Horario de mañana:' (two text inputs for 'Días' and 'Horario' with '+' and '-' buttons), 'Horario tarde:' (two text inputs for 'Días' and 'Horario' with '+' and '-' buttons), and 'Proxima apertura:' (checkbox checked). At the bottom right are two buttons: 'Enviar' (with a red double arrow icon) and 'Cancelar' (with a red double arrow icon).

FIGURA 50 Gestor: Nuevo restaurante

La sección es un formulario html, el cual consta de todos los campos necesarios para crear un restaurante en el sistema. Existen dos casos, creación de un restaurante abierto y la creación de un restaurante que aun no ha sido abierto.

La diferencia entre un restaurante abierto y los que no están abiertos radica en los campos que se deben rellenar. Para indicar al gestor que un restaurante aun no esta abierto se debe marcar la casilla contigua a la frase “Próxima apertura”. En caso de que un restaurante sea próxima apertura solo son necesarios los campos: nombre del restaurante y la ciudad.

Para crear un restaurante abierto es necesario rellenar casi todos los campos. A continuación se va a pasar a explicar que es cada campo:

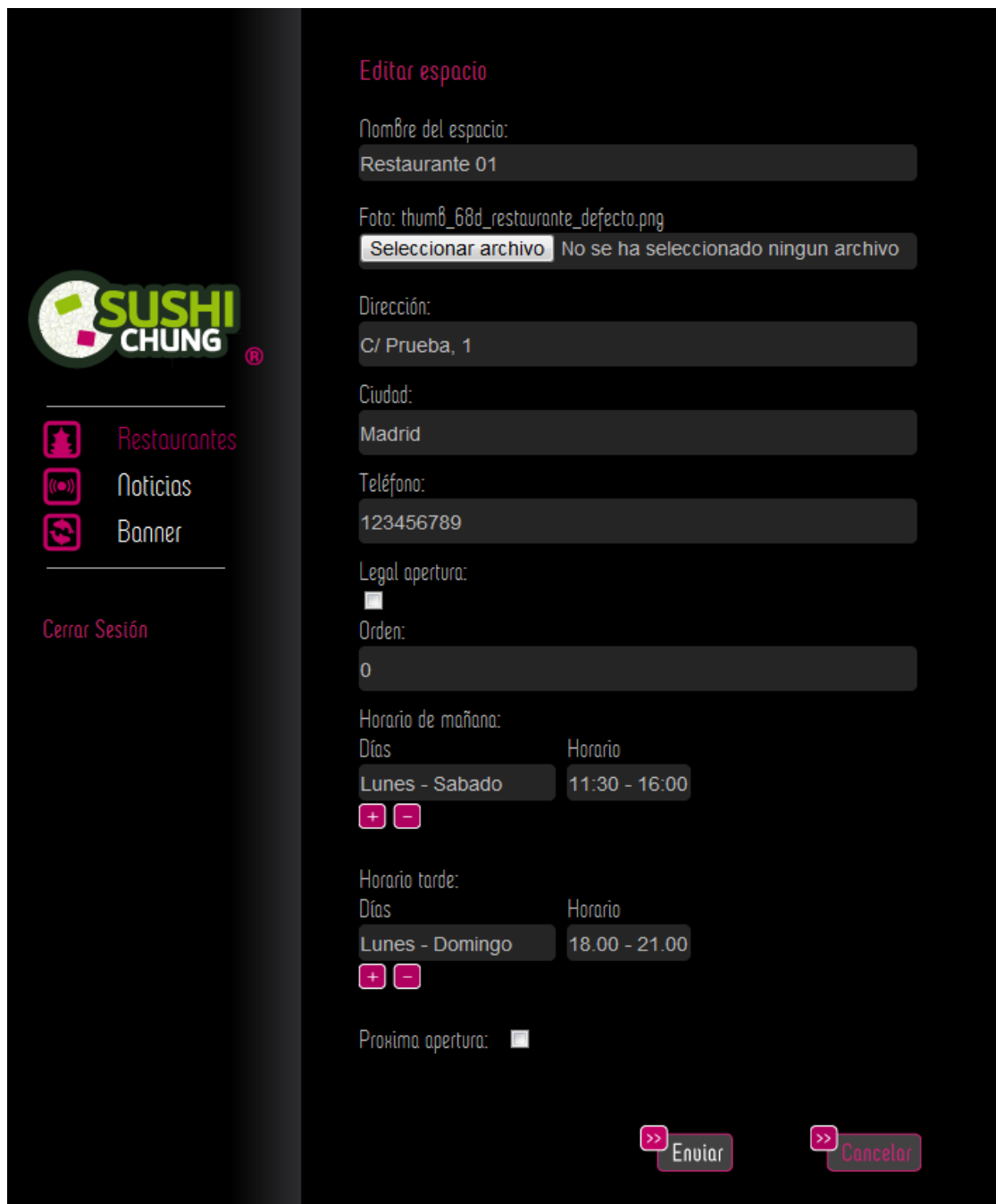
- Nombre del restaurante: El campo es obligatorio. Este campo es el nombre dado al restaurante. En la página principal y en el gestor de restaurantes se usa para las entradas de los índices correspondientes a los restaurantes respectivamente.
- Foto: Este campo no es obligatorio. La foto debe ser una foto del local, la idea es atraer al usuario al restaurante, por lo tanto ha de ser una foto que llame al usuario a ir al restaurante. En caso de que no se disponga de una foto el sistema introduce en este campo la foto por defecto para todos los restaurantes.
- Dirección: El campo es obligatorio. Este campo corresponde a la dirección donde se encuentra el restaurante.
- Ciudad: Este campo es obligatorio, indica la ciudad donde se encuentra el restaurante.
- Teléfono: El campo es obligatorio, indica el teléfono del restaurante. El campo ha de ser un teléfono valido, por lo tanto debe tener de 1 a 11 dígitos.
- Legal apertura: No es obligatorio, sirve para indicar al sistema si al mostrar la información del restaurante tiene que parecer un pequeño texto legal debajo indicando que el restaurante se encuentra en un centro comercial y por lo tanto esta sujeto a los horarios y días laborables del centro comercial.

- Orden: No es obligatorio, el campo sirve para posicionar al restaurante en la lista de restaurantes de la página principal. Cuanto mayor es el valor del campo orden mas arriba en la lista de restaurantes de la página principal aparece.
- Horario de mañana: El campo es obligatorio. Consta de dos campos, Días y horario. En los días se deben indicar los días que abre por la mañana en el horario indicado en el campo horario. Debajo del campo Días se encuentra un botón con el símbolo más y al lado un botón con el símbolo menos. Estos botones sirven para poder introducir varios horarios de mañana, para el caso de que los horarios de apertura por la mañana difieran dependiendo del día. Como máximo se permiten 7 horarios por la mañana y como mínimo se ha de rellenar un horario de mañana.
- Horario tarde: El campo es obligatorio. Tiene las mismas características que el horario de mañana. Se debe introducir como mínimo un horario de tarde, y como máximo 7 horarios de tarde (uno por cada día de la semana).
- Próxima apertura: El campo no es obligatorio. La función de este campo es indicar al sistema si el restaurante que se esta creando esta abierto o no.

Una vez el usuario pincha en el enlace enviar, se comprueban que todos los campos sean correctos mediante un función de JavaScript, en caso de que algún campo sea incorrecto se indica al usuario mediante un ventana emergente cual es el campo incorrecto y cuales son las condiciones que debe cumplir el campo para ser aceptado por el sistema.

4.3.3.3- Editar Restaurante

En esta sección del gestor un usuario gestor puede modificar cualquier campo del restaurante seleccionado previamente en la página correspondiente al gestor de restaurantes.



The screenshot shows a web interface for editing a restaurant. On the left is a dark sidebar with the 'SUSHI CHUNG' logo and three menu items: 'Restaurantes' (with a fork and knife icon), 'Noticias' (with a speech bubble icon), and 'Banner' (with a circular arrow icon). Below these is a 'Cerrar Sesión' link. The main area is titled 'Editar espacio' in red. It contains several form fields: 'Nombre del espacio:' with the value 'Restaurante 01'; 'Foto:' with a thumbnail and a 'Seleccionar archivo' button; 'Dirección:' with the value 'C/ Prueba, 1'; 'Ciudad:' with the value 'Madrid'; 'Teléfono:' with the value '123456789'; 'Legal apertura:' with an unchecked checkbox; 'Orden:' with the value '0'; 'Horario de mañana:' with a table showing 'Lunes - Sabado' from '11:30 - 16:00'; 'Horario tarde:' with a table showing 'Lunes - Domingo' from '18.00 - 21.00'; and 'Proxima apertura:' with an unchecked checkbox. At the bottom right are 'Enviar' and 'Cancelar' buttons.

Editar espacio

Nombre del espacio:
Restaurante 01

Foto: thumb_68d_restaurante_defecto.png
 No se ha seleccionado ningun archivo

Dirección:
C/ Prueba, 1

Ciudad:
Madrid

Teléfono:
123456789

Legal apertura:
☐

Orden:
0

Horario de mañana:

Días	Horario
Lunes - Sabado	11:30 - 16:00
<input type="button" value="+"/> <input type="button" value="-"/>	

Horario tarde:

Días	Horario
Lunes - Domingo	18.00 - 21.00
<input type="button" value="+"/> <input type="button" value="-"/>	

Proxima apertura: ☐

FIGURA 51 Gestor: Editar restaurante

Para saber que campos son obligatorios y cuales no el sistema realiza el mismo método de comprobación de campos que en el caso de la creación de un nuevo restaurante, dependiendo también del tipo de restaurante que se esté modificando. En caso de que algún campo obligatorio se encuentre vacío, o no cumpla los requisitos de aceptación del sistema, se indica al usuario mediante una ventana emergente el campo erróneo y cuales son las condiciones para ser aceptado.

Si el usuario desea modificar un restaurante para cambiarlo del estado de restaurante de próxima apertura al estado de restaurante abierto, debe rellenar todos los campos necesarios para la creación de un nuevo restaurante, cumpliendo las mismas condiciones que el caso de la creación de un nuevo restaurante. El usuario además a de indicarlo pulsando la casilla “Próxima apertura”, quedando esta deseleccionada.

4.3.3.4- Gestor Noticias

Se muestra una tabla de tres columnas;

- Primera columna (Noticia): listado de las noticias almacenadas en el sistema con los títulos de las noticias.
- Segunda columna (Acción): La columna contiene dos posibles acciones para cada noticia almacenada en el sistema. Acción Editar, para modificar los datos de la noticia, y borrar, para borrar completamente la noticia del sistema.
- Tercera columna (Noticia destacada): En esta columna el usuario gestor puede seleccionar la noticia destacada, la cual será la noticia mostrada al cargar la página web principal del sistema. En caso de que un usuario gestor cambie la noticia destacada aparece un botón para salvar los cambios.

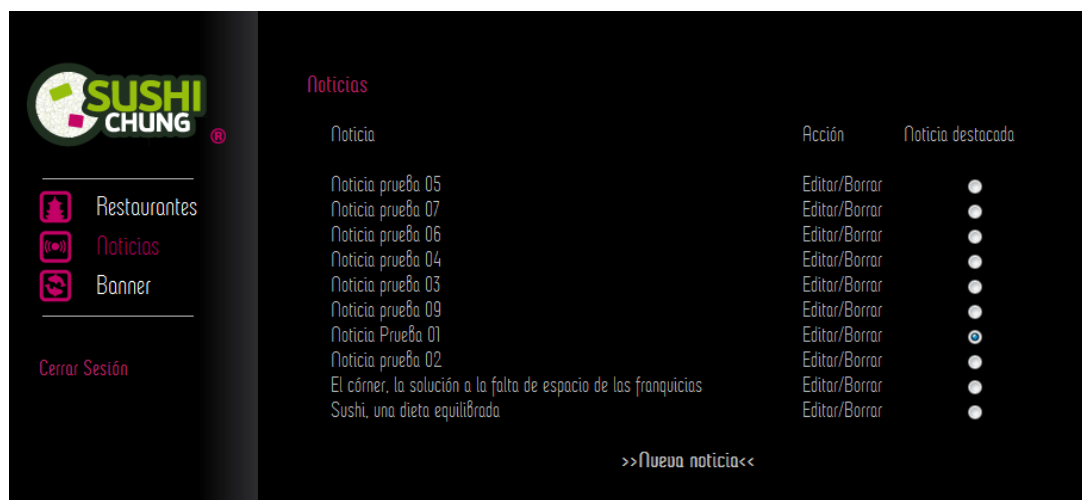


FIGURA 52 Gestor noticias

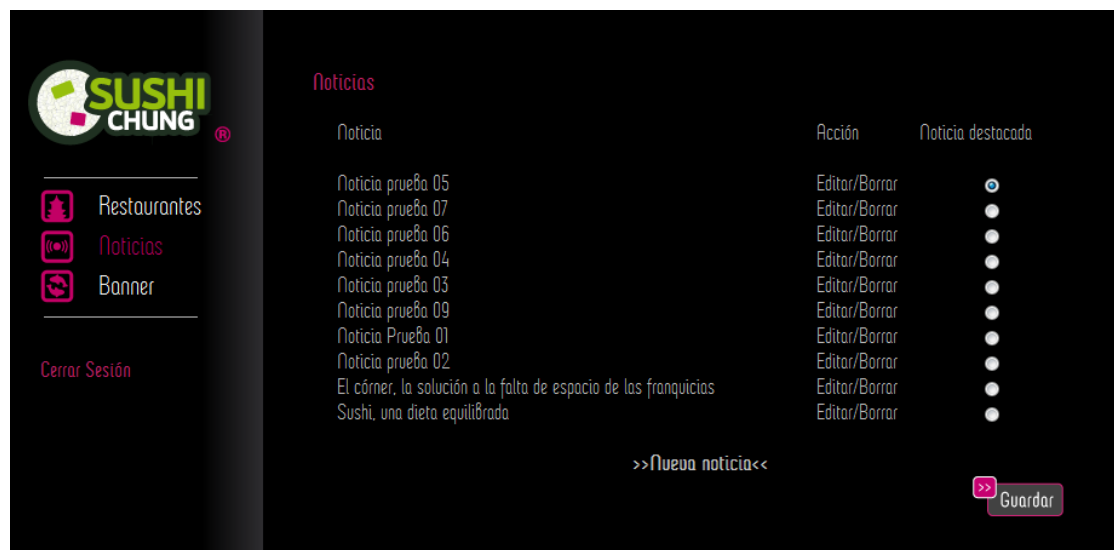


FIGURA 53 Gestor noticias cambio de la noticia destacada

Por ultimo consta del enlace “>>Nuevo noticia<<”, si el usuario pincha dicho enlace este redirige al usuario a la pagina de creación de noticias del gestor.

Cuando el usuario gestor pulsa el enlace borrar se muestra una venta emergente preguntando al usuario si esta seguro de borrar dicha noticia. En caso afirmativo la noticia se borra del sistema y se vuelve a cargar el listado de noticias. En caso contrario no se realizan más acciones.

Si el usuario gestor intenta borrar la noticia destacada el sistema muestra en una venta emergente un mensaje de error en el que se indica que la noticia destacada no puede ser borrada, ya que el sistema requiere una noticia destacada para ser mostrada en la carga de la pagina principal. Si es necesario borrar la noticia destacada el usuario gestor deberá indicar cual va a ser la futura noticia destacada, marcarla, guardar los resultados y a continuación borrar la noticia.

4.3.3.5- Nueva Noticia

La sección crea nuevas noticias en el sistema. Para ello se ha creado un formulario que ha de cumplimentar el usuario gestor para crear una noticia en el sistema. No todos los campos son obligatorios, a continuación se detalla cada uno de ellos:

- Titular noticia: Es el titular de noticia, se utilizara en los índices para referenciar dicha noticia. Es un campo obligatorio.
- Fuente de la noticia: Origen de la noticia, es un campo obligatorio. Es posible que la fuente de la noticia sea el propio usuario gestor, en ese caso el usuario puede introducir su nombre o el de la franquicia.
- Fecha: fecha de publicación de la noticia. El campo es obligatorio.
- Contenido: Cuerpo de la noticia. El campo es obligatorio.
- Foto: Es posible que la noticia contenga una foto, en ese caso el usuario ha de subir la foto al servidor a través de este campo, el cual no es obligatorio. La foto será mostrada en la página principal en la parte derecha de la noticia.
- Noticia destacada: su funcionalidad es indicar al sistema si la noticia que se va a crear a de pasar a ser la noticia destacada, remplazando a la actual noticia destacada del sistema. El campo no es obligatorio, en caso de no estar marcado en la creación de la noticia el sistema interpretara que noticia a crear no es la noticia destacada.
- Link: en caso de que la noticia sea una noticia externa el usuario puede introducir en este campo la dirección de la noticia original. Si el campo sea rellenado se crea un enlace en la página principal al final de noticia el cual abre en el explorador de internet empleado una nueva ventana con la dirección introducida este campo. El campo no es obligatorio.

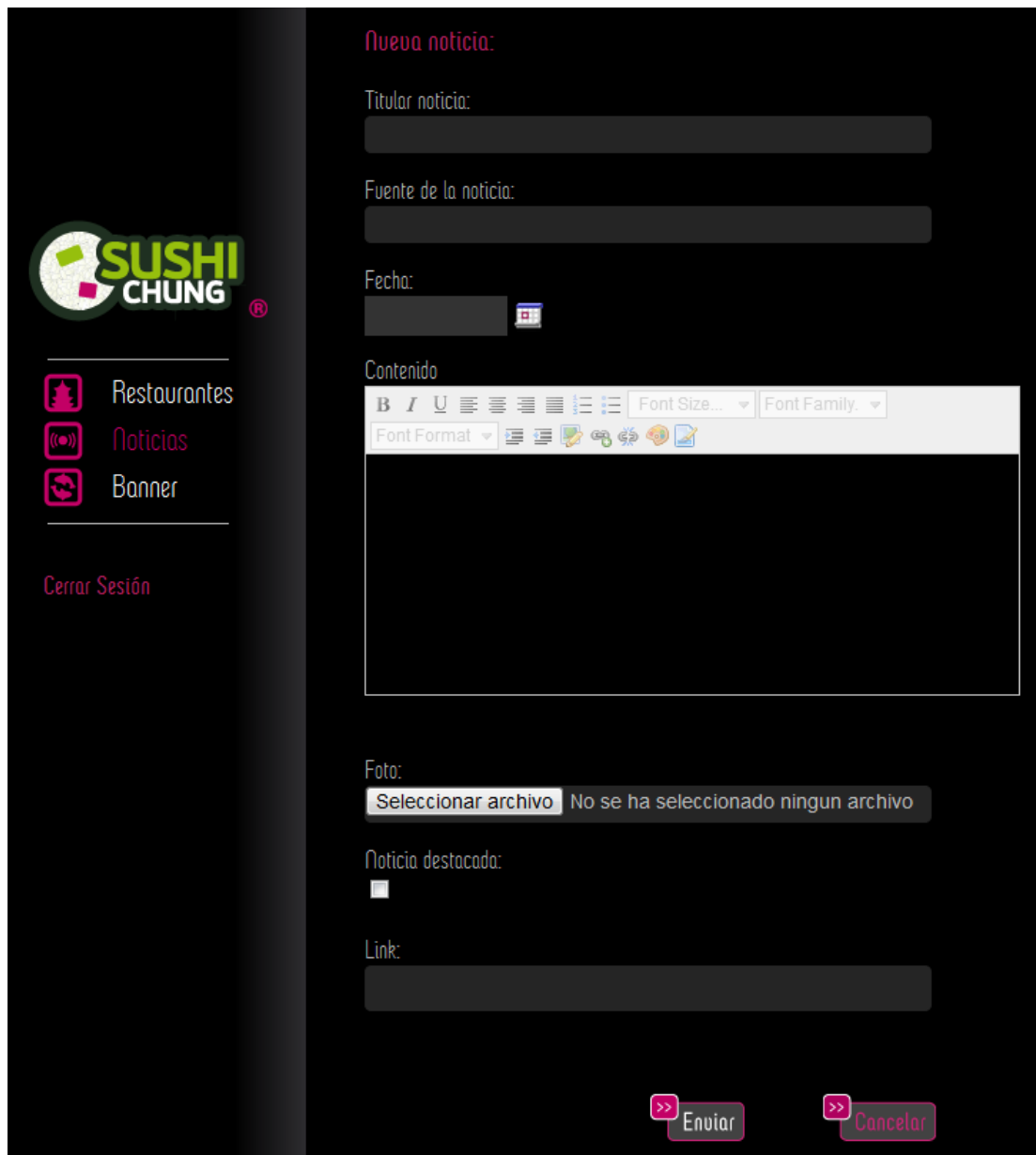


FIGURA 54 Gestor: Nueva noticia


Una vez se haya creado la noticia sin errores, se muestra un mensaje indicando que la noticia ha sido creada satisfactoriamente.

4.3.3.6- Editar Noticia

Para editar noticias se ha creado esta página en el gestor, la cual consta de un formulario con todos los posibles campos que puede tener una noticia.

Los campos aparecen rellenos con los datos almacenados en la base de datos. El usuario puede modificar cualquiera de ellos, con ciertas restricciones:

- Titular noticia: El campo no puede estar vacío y puede albergar como mucho 200 caracteres.
- Fuente de la noticia: no existe ninguna restricción en este campo.
- Fecha: La fecha es obligatoria, por lo tanto el usuario si desea modificar la fecha de publicación de la noticia ha de marcar la fecha en el calendario emergente.
- Contenido: El campo a de contener al menos un carácter.
- Foto: el campo puede estar vacío. El único requisito de este campo es que el archivo subido como foto a de ser una imagen con una extensión de imagen reconocida.
- Noticia destacada: no existe ninguna restricción en este campo.
- Link: no existe ninguna restricción en este campo.



- Restaurantes
- Noticias
- Banner

Cerrar Sesión

Editar noticia

Titular noticia:
Noticia prueba 05

Fuente de la noticia:
loremipsum.net/

Fecha:
2012-05-22

Contenido

B *I* U [List Icons] Font Size... Font Family.

Font Format [List Icons]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius. Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum.

Foto:
[Seleccionar archivo] No se ha...archivo

Link:
http://www.loremipsum.net/

[Enviar] [Cancelar]

FIGURA 55 Gestor: Editar noticia

4.3.3.7- Gestor Banner

Es esta pagina del gestor se encuentra una tabla con dos columnas, la columna de Banner que contiene el listado de todas las diapositivas del banner, a las que se hace referencia mediante su id, y la columna acción, en donde se indican las dos posibles acciones que se pueden llevar a cabo en las diapositivas del banner, dichas acciones son Editar y Borrar.

Al final de la tabla se encuentra una fila que ocupa dos columnas. Existe un enlace denominado ">>Nueva diapositiva<<", el cual redirige al usuario a la pagina de creación de diapositivas del banner.

Al final del contenido de la página se halla el intervalo de tiempo entre imágenes. Este intervalo representa al tiempo en el que se muestran las diapositivas del banner, o lo que es lo mismo el tiempo que transcurre del paso de una diapositiva del banner a otra. Se el usuario modifica este campo, aparece un botón para salvar los resultados.



FIGURA 56 Gestor Banner

4.3.3.8- Nueva diapositiva

Página de creación de diapositivas del banner. El contenido de la página cuenta con un formulario con todos los campos necesarios para la creación de la diapositiva del banner. Todos los campos son obligatorios. A continuación se detalla la funcionalidad de cada uno de ellos:

- Contenido: El campo representa el texto que contiene la diapositiva del banner.
- Posición del contenido: Su funcionalidad es indicar al sistema en que lugar se ha de posicionar el texto del banner. Existen dos posibilidades; izquierda y derecha. La opción izquierda sitúa al texto en la mitad izquierda del banner, y la opción derecha sitúa el texto en la mitad derecha del banner.
- Fondo: imagen de fondo de la diapositiva a crear.

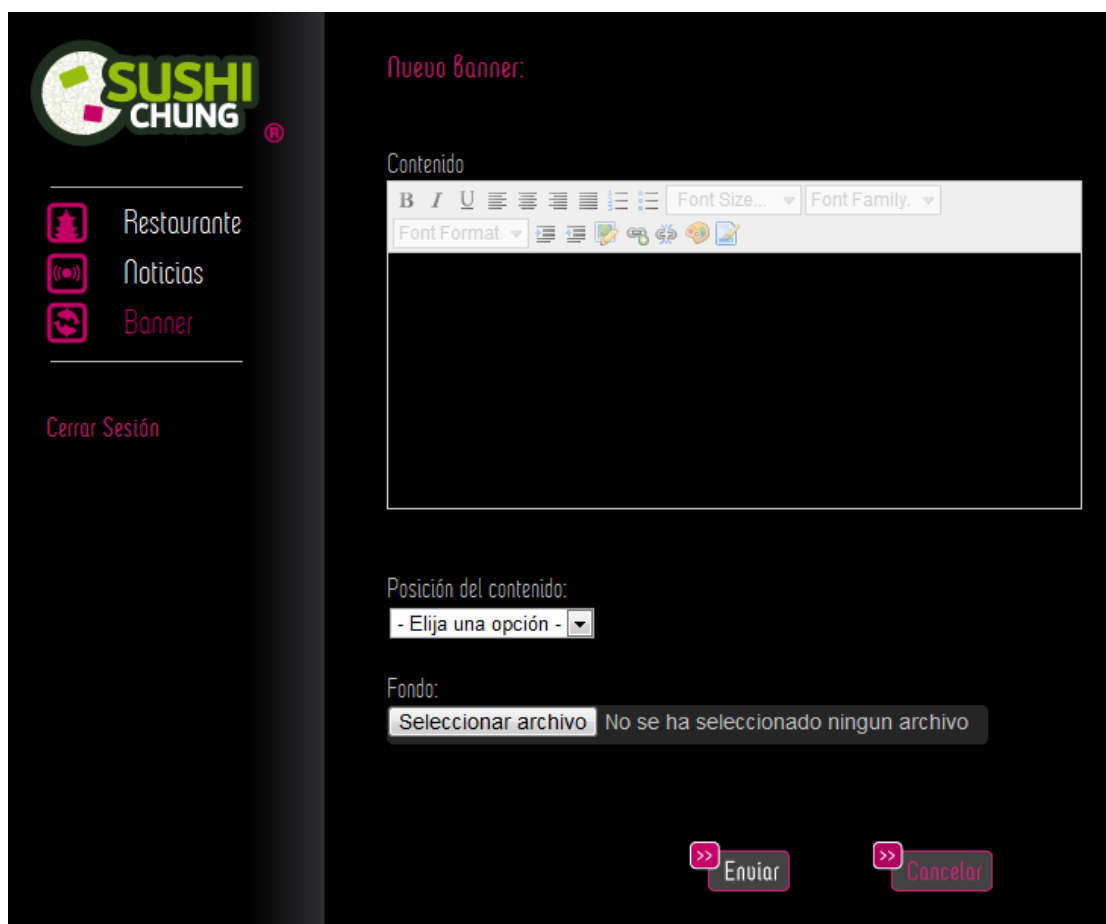


FIGURA 57 Gestor: Nueva diapositiva para el banner



En caso de que un campo sea erróneo para el sistema, este muestra una ventana emergente indicando el campo erróneo y que restricciones a de cumplir.

4.3.3.9- Editar diapositiva

Para editar una diapositiva del banner el usuario gestor a de pulsar el enlace editar correspondiente a la diapositiva que desea modificar, desde el gestor del banner.

La página edición de diapositivas cuenta con un formulario, igual al que se encuentra en la página de creación de nuevas diapositivas del banner, pero en este caso el formulario aparece con todos los campos rellenos con los datos almacenado en la capa de datos (Base de datos).

Todos los campos son obligatorios, y por lo tanto ninguno puede estar vacío, pueden ser modificados pero no se puede eliminar por completo el contenido de ninguno de ellos.



FIGURA 58 Gestor: Editar diapositiva para el banner

4.4.- Capa de Negocio

4.4.1.- Introducción

Para explicar la capa de negocio se tiene que explicar primero sección por sección como funciona en la página web principal de la aplicación y como se comunica con la capa de presentación y con la capa de datos.

Algunas de las secciones de la pagina principal no contiene acciones para la capa de negocio, y por lo tanto no serán vistas en esta parte de la memoria.

Una vez terminada la explicación de las secciones de la página principal se realizara la explicación del gestor, empleado el mismo método, se irán presentando una a una cada una de las secciones de las que esta compuesto, explicando como funciona la capa de negocio, y como esta se comunica con las demás capas.

La capa de negocio esta formada principalmente por código PHP, es inevitable que la explicación de esta capa contenga algo de código PHP y HTML. PHP es el que interactúa con la capa de datos y es el que genera parte del código HTML en función de diversos factores que se van a explicar en detalle sección por sección.

Hay que tener en cuenta que en la capa de presentación se encuentra el código JavaScript y jQuery. Podría parecer que esta parte del código debería estar en la capa de negocio, pero no es así, debido a que la capa de negocio es la capa que recibe las peticiones del usuario y envían las respuestas tras el proceso, por lo tanto el código JavaScript y jQuery nunca podrá formar parte de la capa de negocio, si no de la capa de presentación, en donde se encuentra el código que es ejecutado por el usuario, en este caso entra el código JavaScript y jQuery que es ejecutado por el ordenador del usuario a través del explorador, es decir la interpretación de este código corre a cargo del computador de usuario.

En cada sección se explica como funciona la capa de negocio, y a continuación el pseudocódigo de las funciones que realiza la capa de negocio. Tanto la explicación como el pseudocódigo son un resumen de la programación realizada.

4.4.2- Página principal

En la página principal la capa de negocio realiza las funciones de acceso a la base de datos, obtiene información de diversas secciones. También tiene a cargo el envío del e-mail de consultas de los usuarios cuando estos rellenan el formulario que se encuentra en la sección contacto.

Para solicitar la información a la base de datos se ha creado un documento PHP (func.php), el cual contiene todas las funciones creadas para consultar, insertar, modificar y eliminar contenido de la base de datos. Además contiene la información de acceso a la base de datos para poder realizar todas las acciones anteriormente mencionadas.

Pseudocódigo:

Cargar librerías;

SI se ha recibido el formulario para el envío de consultas {
 Carga de variables del formulario;

 Carga de datos para el envío del mail;
 Envío (mail de consulta del usuario);
}

Consulta Recuperar tiempo de la tabla tb_tiempos donde
nombre="banner_1";

SI se ha recuperado tiempo
 Variable tiempo banner = tiempo recuperado;
SI NO
 Tiempo_banner = 500;

Nota: Tiempo_banner es la variable donde se almacena el tiempo que va transcurrir en el paso de las diapositivas del banner.

4.4.2.1- Sección Banner

La capa de negocio solicita a la base de datos toda la información almacenada en la tabla “tb_banner” mediante una consulta PHP, la cual devuelve el resultado en un matriz de datos, en donde se encuentran todos los registros de la tabla “tb_banner”.

Una vez se obtiene el resultado de la consulta, mediante PHP se escribe el código HTML que corresponde con al código del banner, en donde se crea un índice y cada registro del índice pasa a ser una diapositiva del banner, tal y como se ha explicado en la capa de presentación.

Para poder realizar dicho índice, el cual mas tarde pasara a ser el banner de la página web, se ha creado un bucle, el cual se repite hasta leer todos los registros almacenados en la matriz devuelta por la consulta anteriormente mencionada. En cada repetición del bucle se genera una dispositiva del banner, manteniendo del formato de lista en HTML, generando el código tanto para la imagen de fondo, como para texto que contiene.

Dependiendo de si la diapositiva del banner tiene el texto en la lado derecho o en el lado izquierda de la diapositiva se la da una clase u otra, existiendo por lo tanto dos clases para el posicionamiento del texto dentro de cada diapositiva del banner, de esta manera se posiciona el texto en cada diapositiva del banner de un modo muy sencillo.

Pseudocódigo:

```
Array datos_banner = Consulta recuperar los datos de la tabla “tb_banner”;
```

```
MIENTRAS existan datos en el array datos_banner {
```

```
    SI datos_banner [posicion] = izq
```

```
        i = 2;
```

```
    SI NO i = 1;
```

```
    imprimir imagen con la dirección datos_banner [“imagen”];
```

```
    imprimir datos_banner [“texto”] con la clase texto_diaposisitiva_i;
```

```
}
```

Nota: Por cada repetición de bucle PHP pasa a la siguiente posición del array.

4.4.2.2- Sección Restaurantes

Antes de mostrar la página principal se realizan las consultas a la base de datos, tanto en esta sección, como en la sección banner y la sección noticias, y se genera el código necesario para cada una de estas secciones.

En este caso se realizan dos consultas a la base de datos, la primera consulta tiene como objetivo recoger toda la información de los registros de la tabla restaurantes en donde los restaurantes han de ser restaurantes abiertos.

Una vez obtenido el resultado de la consulta en un matriz de datos se pasa a generar el código en HTML de los índices de los restaurantes empleado PHP. Dicho código HTML esta formado el índice de restaurantes abiertos, en donde se crea una entrada del índice por cada restaurante abierto devuelto por la consulta anterior. Cada entrada del índice hace referencia a cada restaurante a través del nombre del restaurante escritor en el gestor.

Después se pasa a crear el índice de restaurantes que aun no están abiertos, empleado el mismo método, se realiza una consulta a la base de datos solicitando toda la información almacenada en la base de datos de los restaurantes que aun no han abierto. Con el resultado de la consulta se pasa a escribir el código HTML correspondiente al índice de restaurantes que aun no han abierto, o próximas aperturas, del mismo modo que en el índice anterior, por cada restaurante obtenido en la consulta se crea una entrada en el índice con su nombre.

Una vez finalizada la creación de los índices de restaurantes se pasa a generar el código para mostrar la información de los restaurantes abiertos. Como la información a mostrar tiene siempre el mismo formato se ha creado un bucle, el cual se repite hasta acabar todos los registros de la tabla restaurantes obtenidos por la consulta. En cada repetición del bucle se imprime primero el nombre del restaurante, seguidamente la foto del restaurante y un marco que la rodea. Es importante saber que el tamaño de las fotos de los restaurantes es siempre la misma, en caso de que el usuario suba a través del gestor una foto con un tamaño menor o mayor se genera una imagen por PHP del tamaño correspondiente, se realizara una explicación mas detallada en la sección del gestor que corresponda.

Una vez generada la imagen se pasa a generar el código de la información del restaurante; dirección, teléfono, horario de mañana, horario de tarde...etc.

El orden el que aparecen los restaurantes es el orden depende del valor de la variable orden que contengan en la base de datos.

Pseudocódigo:

Array datos_restaurantes = Consulta recuperar los datos de restaurantes abiertos de la tabla “tb_restaurntes”;

i = 1;

MIENTRAS existan datos en el array datos_restaurantes {

 SI i=1 {

 Imprimir enlace del índice con el texto datos_restaurantes (“entrada_indice”) y la clase activo;

 SI NO {

 Imprimir enlace del índice con el texto datos_restaurantes (“entrada_indice”);
 }

 }

}

Array datos_restaurantes = Consulta recuperar los datos de restaurantes NO abiertos de la tabla “tb_restaurntes”;

MIENTRAS existan datos en el array datos_restaurantes {

 Imprimir enlace del índice con el texto datos_restaurantes (“entrada_indice”);

}

Array datos_restaurantes = Consulta recuperar los datos todo los restaurantes de la tabla “tb_restaurntes”;

MIENTRAS existan datos en el array datos_restaurantes {

 Imprimir restaurante;

}

4.4.2.3- Sección Noticias

Se realizan cuatro consultas a la base de datos para generar el código de los índices y noticias de la sección.

Primero se genera el código de los índices, el índice de las noticias de este mes, y el índice de las noticias de meses anteriores. Para ello se realizan dos consultas a la base de datos, en la primera la capa de negocio realiza una consulta a la base de datos para obtener todas las noticias del mes actual ordenadas cronológicamente, una vez se obtiene el resultado de la consulta este se pasa a una matriz de datos, que es leída mediante PHP.

Dicha matriz de datos contiene los datos de todos los registros de las noticias de la base de datos con fecha de este mes. Se genera el código del índice y se crea una entrada del índice por cada registro obtenido en la matriz de datos devuelta por la consulta anterior.

A continuación se realiza el mismo proceso para las noticias de meses pasados, mediante PHP se hace una consulta a la base de datos para recuperar la información de todas las noticias de meses anteriores y se genera el código HTML del índice, en donde cada entrada del índice corresponde al título de una noticia.

Cada entrada del índice es un enlace que contiene función JavaScript, al invocarla pulsando un enlace del índice hace visible la noticia a la que corresponde y oculta el resto de noticias.

Una vez creados los índices se pasa a crear el código HTML de las noticias, para ello la capa de negocio primero hace una consulta a la base de datos solicitando la información de la noticia destacada y se genera el código HTML de la noticia destacada. Para diferenciar las noticias se usa el id, las noticias poseen el mismo comienzo de id pero con diferente número. El número de cada noticia es generado con el orden en el que fueron creadas, es decir la segunda noticia para la que se genera su código HTML poseerá el id “noticia_2”. Hay que tener en cuenta que el orden en el que se imprimen las noticias, excepto la noticia destacada, sigue un orden cronológico, del más actual al menos actual.

Para generar las noticias no destacadas se realiza otra consulta a la base de datos solicitando la información de todos los registros de la tabla “tb_noticias” en donde la variable “destacado” sea igual a 0, o lo que es lo mismo no sean noticias destacadas.

Para generar el código de las noticias se ha creado una función PHP que se encuentra en el archivo “func.php”, ya que todas las noticias tienen el mismo formato para el código html. Dicha función tiene el nombre “imprimir_noticia”, la cual comprueba si la noticia consta de foto, en caso afirmativo se pasa a comprobar su tamaño mediante código PHP, en caso de que el tamaño de la foto supere un ancho de 130 píxeles, en el código generado para la imagen en HTML se le indica a la foto que ha de tener un ancho de 130 píxeles. Con esto se consigue eliminar los errores ocasionados por los diferentes tamaños de las fotos. La razón de estos errores se debe a que el espacio disponible para mostrar las noticias es siempre el mismo, y por lo tanto es necesario, mantener un tamaño máximo para todas las fotos, y así asegurar un espacio óptimo para el texto de las noticias.

En caso de que la noticia no disponga de foto se emplea el mismo espacio para el texto, de esta manera el texto de todas las noticias se mantiene en la misma posición.

La función “imprimir_noticia” continua generando el código HTML para el contenido de la noticia, en donde se encuentra el título, la fecha, la fuente y el contenido de la noticia. Seguidamente la función comprueba si la noticia posee link, si consta de link se genera el código HTML correspondiente al link. Este código es simplemente un enlace a la dirección del link, en donde dicho enlace se muestra mediante las palabras “leer más” al final de la noticia.

Pseudocódigo:

Id_noticia_destacada = Consulta recuperar noticia destacada en la tabla “tb_noticias”;

Array datos_noticias = Consulta recuperar noticias de este mes de la tabla “tb_noticias”;

```
MIENTRAS existan datos en el Array datos_noticias {  
    SI datos_noticias [“id”] = Id_noticia_destacada {  
        Imprimir enlace del índice con el texto datos_noticias  
        (“entrada_indice”) con la clase activo;  
    } SI NO {  
        Imprimir enlace del índice con el texto datos_noticias  
        (“entrada_indice”);  
    }  
}
```

Array datos_noticias = Consulta recuperar noticias de meses anteriores de la tabla “tb_noticias”;

```
MIENTRAS existan datos en el Array datos_noticias {  
    SI datos_noticias [“id”] = Id_noticia_destacada {  
        Imprimir enlace del índice con el texto datos_noticias  
        (“entrada_indice”) con la clase activo;  
    } SI NO {  
        Imprimir enlace del índice con el texto datos_noticias  
        (“entrada_indice”);  
    }
```

Array datos_noticias = Consulta recuperar noticia destacada en la tabla “tb_noticias”;

Imprimir noticia (datos_noticias);

Array datos_noticias = Consulta recuperar noticia sin noticia destacada en la tabla “tb_noticias”;

```
MIENTRAS existan datos en el Array datos_noticias {  
    Imprimir noticia (datos_noticias);  
}
```

4.4.2.4- Sección Contacto

En esta sección la capa de negocio no realiza ninguna consulta a la base de datos ni genera código HTML, si no que la función de la capa de negocio para esta sección es la de enviar correos con las consultas de los usuarios al e-mail creado para este cometido.

Para que un usuario pueda enviar una consulta a de cumplir los requisitos impuesto pos la capa de presentación, mencionados anteriormente. Una vez la capa de presentación confirma que se han cumplido los requisitos para enviar una consulta realiza la acción “submit” sobre el formulario, empleando el método “POST”, a través del cual se envían las variables en modo oculto a la misma pagina.

Al comienzo de la página web existe un código PHP destinado a la función de envío de consultas. Este código PHP primero comprueba que le haya sido enviado mediante post el formulario para realizar una consulta, en caso de no recibir nada no realiza ninguna acción.

Si este código recibe el formulario de consultas genera un pequeño e-mail con el nombre, el e-mail y la consulta introducidos en el formulario de consultas.

Para realizar en envío se ha empleado la función “mail” de PHP, la cual requiere de un e-mail de destino, el asunto y el contenido del e-mail. El e-mail de destino es un e-mail de gmail, creado para este cometido.

4.4.3- Gestor

El gestor es la parte del sistema mas segura, por ello cuando un usuario intenta acceder al gestor se le exige ingresar su email y contraseña y de este modo se autentifica ante al sistema. Para que el usuario pueda identificarse la capa de negocio realiza una consulta a la base de datos a la tabla “tb_usuarios”, dicha consulta solicita la información del usuario, si existe, con el e-mail y contraseñas introducidos por el usuario.

Si el existe un usuario en la tabla “tb_usuarios” con el e-mail y la contraseña indicados se redirige al usuario a la pagina de gestión de restaurantes, enviando en oculto su e-mail y contraseña. Para enviar los datos de acceso al gestor en modo oculto se envía un formulario con el e-mail y contraseña del usuario empleando el método “POST”.

En todas las páginas del gestor, excepto la página inicial, la capa de negocio exige un nombre de usuario y una contraseña, si no se dispone de ellos se redirige la página web a la pagina de inicio del gestor, obligando al usuario a autentificarse ante el sistema. Para que la capa de negocio pueda leer los datos de acceso de los usuarios en cada pagina del gestor se emplean sentencias PHP, las cuales leen los datos ocultos enviados por los formularios y realizan una consulta a la base de datos solicitando la información del usuario con el e-mail y contraseña introducidos por el usuario.

Pseudocódigo:

Cargar librerías;

```
SI se a recibido el formulario para logarse {  
    Validar (formulario (“usuario”), formulario (“contraseña”));  
    Si NO es valido {imprimir mensaje de error;}  
}
```

```
/*
```

Existe un formulario oculto con las variables ocultas mail de usuario y contraseña. Si se ha recibido el formulario para logarse la capa de negocio imprime las variables recibidas en las variables ocultas del formulario oculto, según correspondan.

```
*/
```

```
SI es valido {  
    Imprimir sentencia javascript, la cual ejecuta submit en el formulario  
    oculto;  
}
```

Nota: La función validar realiza una consulta a la base de datos solicitando el registro de la tabla “tb_usuarios” con el mismo nombre y la misma contraseña que los enviados en el formulario para logarse. Antes de realizar esta consulta la función invoca la función anti_injection, dando como dato de entrada a esta función el nombre y la contraseña enviados en el formulario para logarse.

La función anti_injection elimina todas las posibles sentencias SQL de la variable de entrada dada y la devuelve después de la eliminación de dichas sentencias.

4.4.3.1- Gestor Restaurantes

Se crean dos índices con los nombres de los restaurantes, el primer índice esta formado por los restaurantes abiertos y el segundo índice con los restaurantes que aun no han sido abiertos y son considerados como próximas aperturas.

Para poder crear los índices es necesario realizar dos consultas a la base de datos, una para cada índice.

Primero se realiza una consulta a la base de datos para que devuelva todos los restaurantes abiertos almacenados en la base de datos. El resultado de la consulta se almacena en una matriz de datos, la cual contiene todos los registros de los restaurantes abiertos encontrados en la base de datos.

Se genera, mediante sentencias PHP, el código HTML de la tabla que alberga los índices, comenzando por la primera fila, en donde se indica al usuario que es cada columna de la tabla. A continuación se pasa a crear el índice de los restaurantes abiertos, en donde cada entrada del índice corresponde al nombre de cada restaurante abierto.

En la columna de acciones (Editar/Borrar), se crea para cada acción un enlace, el cual contiene una sentencia JavaScript que invoca una función JavaScript. Se han creado dos funciones JavaScript, una para borrar y otra para editar registros de restaurantes. Ambas funciones modifican el valor de la variable oculta que contiene el id del registro de la base de datos correspondiente al restaurante sobre el que se quiere realizar la acción. Una vez modificada dicha variable se ejecuta la sentencia “submit” de uno de los formularios, dependiendo de la acción a realizar.

Si la acción a realizar es borrar ejecuta la sentencia “submit” del formulario “frm_borrar” el cual recarga la página actual, enviando mediante el método “POST” los datos del restaurante a borrar y los datos de acceso del usuario en modo oculto. Se reciben los datos y se ejecuta una función PHP, la cual elimina de la base de datos el registro de la tabla de restaurantes de la base de datos con el id recibido.

En caso de querer editar los datos de un restaurante se modifica en formulario “frm_editar” el valor de la variable oculta que alberga el id del restaurante. El formulario tiene como acción enviar al usuario a la página de edición de restaurantes. Dicho formulario envía los datos de acceso del usuario y el id del registro del restaurante del cual se quiere modificar los datos.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {

 Validar (formulario (“usuario”), formulario (“contraseña”));

 SI es valido {

 SI se ha recibido el formulario para borrar {

 Borrar_restaurante (id recibido)

 }

 Array restaurantes_abiertos = Consulta recuperar datos de los restaurantes abierto de la tabla tb_restaurantes;

 Array restaurantes_cerrados = Consulta recuperar datos de los restaurantes que aun no estan abiertos de la tabla tb_restaurantes;

 MIENTRAS existan datos en el Array restaurantes_abiertos {

 Imprimir restaurantes_abiertos (“entrada_indice”);

 Imprimir enlace que llama a la función editar (restaurantes_abiertos (“id”));

 Imprimir enlace que llama a la función borrar (restaurantes_abiertos (“id”));

 }

 MIENTRAS existan datos en el Array restaurantes_cerrados {

 Imprimir restaurantes_abiertos (“entrada_indice”);

 Imprimir enlace que llama a la función editar (restaurantes_abiertos (“id”));

 Imprimir enlace que llama a la función borrar (restaurantes_abiertos (“id”));

 }

 } SI NO {redirigir a la página de inicio del gestor;}

} SI NO {redirigir a la página de inicio del gestor;}

4.4.3.2- Nuevo Restaurante

Como en el resto de paginas del gestor, primero se solicita el e-mail y contraseña del usuario, se realiza una consulta a la base de datos solicitando el usuario con dichos datos, en caso de existir se mantiene al usuario en la página actual, encaso contrario se redirige al usuario ala pagina principal del gestor par que se autentifique de nuevo. No se realizan mas consultas.

Una vez se ha completado el formulario para crear un nuevo restaurante, y se han cumplido los requisitos exigidos para cada variable de un registro de un restaurante, anteriormente mencionados, se pasa a realizar un “submit” del formulario de creación de un nuevo restaurante. Dicho formulario envía los datos de acceso del usuario en modo oculto, y los datos del nuevo restaurante. Mediante sentencias PHP se comprueba que el usuario es un usuario valido para el sistema y se pasa a comprobar si se han enviado todos los datos para la creación de un restaurante. Se comprueba uno por uno cada dato, en caso de ser un campo obligatorio se comprueba que el campo no este vacío y no supera el limite permitido.

Para todos los campos, tanto obligatorios como no, se comprueba si contienen algún dato, en caso tener datos se comprueba que el contenido no contenga sentencias SQL. De esta manera se evita un gran problema de seguridad de las páginas web, SQL Injection o Inyección SQL. En caso de error no se crea el registro.

En cuanto al horario, un restaurante puede tener varios horarios, tanto de mañana como de tarde. Todos los horarios de mañana y de tarde son enviados a la capa de datos en una sola línea, una línea para los horarios de mañana y otra para los horarios de tarde.

Para poder generar dicha línea se envía con el formulario de creación la cantidad de horarios de tarde y de mañana que ha ingresado el usuario en dos variables, una para los horarios de mañana y otra para los horarios de tarde. Es decir que cada vez que el usuario presiona el botón más, o menos, de uno de los horarios se modifica una de las variables creadas para almacenar el total de horarios.

Los horarios, tanto de mañana como de tarde, tiene un máximo de 7 horarios diferentes, cada uno de esos horarios es una variable del formulario. Todas las variables de horarios que hayan sido rellenadas se juntan en una sola línea, separando cada horario por el símbolo “\$”, el día y la horas se separan con el símbolo “#”. Para ello se creado un bucle, el cual se repite tantas veces como indique la variable de total de horarios, de mañana o de tarde en función del horario que se esté procesando. Por cada repetición del bucle se escribe al comienzo el símbolo “\$”, después los días, seguidos del símbolo “#” y a continuación el horario introducido para esos días.

Con este método se almacenan todos los horarios de los restaurantes en el mismo registro del restaurante, evitando la creación de otra tabla en la base de datos, y como consecuencia evitando más accesos.

Una vez generados las líneas a ingresar de los horarios de mañana y de tarde en la base de datos se siguen comprobando algunas variables hasta llegar al campo foto. El campo foto contiene el fichero a subir al servidor y el nombre del fichero. Para poder enviar fotos con el mismo nombre se ha recurrido a un método muy sencillo, se modifica el nombre de la foto generando 3 número aleatorios con el algoritmo criptográfico MD5. Estos tres números aleatorios se introducen al comienzo del nombre de la foto, añadiendo entre ambos el símbolo “_”.

Una vez generado el nombre de la foto se sube la foto con ese nombre a la carpeta “temp” de la carpeta donde están todas las imágenes en el servidor. A continuación se pasa a crear una copia de foto con el tamaño óptimo para el espacio asignado a las fotos de restaurantes en la página principal, para ello se invoca a la función “createthumb”.

La función “createthumb” se encuentra en el archivo “func.php” y su cometido es crear una copia de la imagen original con el tamaño establecido en su invocación. Para ello primero comprueba que la foto original se ha creado en el servidor, después comprueba la extensión del archivo, para comprobar que el archivo se trata de una foto con una extensión reconocida. Si todo es correcto se pasa a comprobar el tamaño de la foto. Si el tamaño, tanto del ancho como del alto, no corresponde con el tamaño dado para las fotos de restaurantes pasa a calcular el nuevo tamaño para el ancho y alto de la foto. Ya se dispone del nuevo tamaño, y se sabe que el archivo tiene un formato de imagen reconocido, se pasa a generar la nueva foto. Las extensiones reconocidas por la función son PNG, GIF, JPG y JPEG.

Con la foto generada correctamente con el tamaño optimo, y si no ha habido errores, se inserta el nuevo registro en la base de datos.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {

 Consulta Validar (formulario (“usuario”), formulario
 (“contraseña”));

 SI es valido {

 SI se ha recibido form_nuevo /*el formulario para crear un
 nuevo restaurante*/ {

 SI (longitud (form_nuevo (“entrada_indice”)>0 Y
 longitud (form_nuevo (“entrada_indice”) <=200 Y
 Validar (form_nuevo (“entrada_indice”))) {
 Datos_nuevo[(“entrada_indice”)] =
 form_nuevo (“entrada_indice”);

 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“dirección”)>0 Y longitud
 (form_nuevo (“dirección”) <=200 Y Validar
 (form_nuevo (“dirección”))) {
 Datos_nuevo[(“dirección”)] = form_nuevo
 (“dirección”);

 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“ciudad”)>0 Y longitud
 (form_nuevo (“ciudad”) <=150 Y Validar (form_nuevo
 (“ciudad”))) {
 Datos_nuevo[(“ciudad”)] = form_nuevo
 (“ciudad”);

 } SI NO {
 Error = verdadero;
 }

```
SI (longitud (form_nuevo ("telefono"))>0 Y longitud
(form_nuevo ("telefono") <=12 Y Validar (form_nuevo
("telefono"))) {
    Datos_nuevo["telefono"] = form_nuevo
("telefono");
} SI NO {
    Error = verdadero;
}

DESDE x = 1 HASTA x <= form_nuevo
("total_mañana") {
    SI x != 1 {Horario_mañana = Horario_mañana +
"$"}
    SI (Valido (form_nuevo ("horario_mañana"+x+"_dia"))) Y form_nuevo
("horario_mañana"+x+"_dia") <=150 Y (form_nuevo
("horario_mañana"+x+"_hora"))) Y form_nuevo
("horario_mañana"+x+"_hora") <=150)) {
        Horario_mañana = Horario_mañana +
form_nuevo ("horario_mañana"+x+"_dia") + "#" +
form_nuevo ("horario_mañana"+x+"_hora");
    }
}

DESDE x = 1 HASTA x <= form_nuevo ("total_tarde")
{
    SI x != 1 {Horario_tarde = Horario_tarde + "$"}
    SI (Valido (form_nuevo ("horario_
tarde"+x+"_dia"))) Y form_nuevo ("horario_
tarde"+x+"_dia") <=150 Y (form_nuevo ("horario_
tarde"+x+"_hora"))) Y form_nuevo ("horario_
tarde"+x+"_hora") <=150)) {
        Horario_tarde = Horario_tarde +
form_nuevo ("horario_tarde"+x+"_dia") + "#" +
form_nuevo ("horario_tarde"+x+"_hora");
    }
}

SI (longitud (form_nuevo ("primero"))>0 Y longitud
(form_nuevo ("primero") <=3 Y Validar (form_nuevo
("primero"))) {
    Datos_nuevo["primero"] = form_nuevo
```

```
        ("primero");
    } SI NO {
        Error = verdadero;
    }

    SI (form_nuevo ("foto")) {
        Datos_nuevo["foto"] = form_nuevo
        ("foto");
    } SI NO {
        Datos_nuevo["foto"] = foto_porvisonal;

    SI (form_nuevo ("foto")) {
        Prefijo = generar_tres_numero_con_MD5;
        Direccion_foto = prefijo + "_" + form_nuevo
        ("foto").nombre;

        SI subir_foto (form_nuevo ("foto"),
        direccion_foto){
            Resultado = Crear_thumb (Direccion_foto);
            SI (resultado = 1) {
                Mensaje = "La imagen" +
                form_nuevo ("foto").nombre + "a sido creada
                correctamente";
            } SI NO {
                Mensaje = "La imagen es incorrecta";
                Error_foto = verdad;
            }
        } SI NO {
            Mensaje = "Ha ocurrido un error con el servidor";
            Error_foto = verdad;
        }

    SI (NO error Y NO error_foto) {
        Crear registró en la tabla "tb_restaurantes" con los datos
        procesados;
        Mensaje =" El restaurante ha sido creado
        correctamente";
    }
    SI (mensaje) {imprimir mensaje};

    } SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

Nota: Para cada sección del índice se ha creado un formulario, cada uno tiene la acción de redirigir al usuario a una sección. Cada uno de estos formularios tiene como variables ocultas el nombre y la contraseña del usuario. De esta manera cada vez que el usuario quiera ir a otra página del gestor estos datos serán enviados.

Después de comprobar que los datos para logarse el usuario son correctos se imprimen ambas variables en todos los formularios descritos.

4.4.3.3- Editar Restaurante

Al acceder a la página de edición de restaurantes la capa de negocio, empleando PHP, solicita el e-mail del usuario y la contraseña con la que fue logado. Si no se dispone de estos datos se redirige a la pagina de inicio del gestor. Si se disponen de los datos se ejecuta una consulta a la base de datos buscando un usuario con el e-mail y contraseña recibidos, si no existe se redirige a la pagina de inicio del gestor. Se comprueba si se ha recibido el formulario para la edición del restaurante, en este caso no se ha recibido y se pasa a la siguiente acción.

Para acceder a esta pagina se tiene que pulsar en enlace “editar” de uno de los restaurantes en la pagina de gestión de restaurantes, y por lo tanto se envía en modo oculto el id del restaurante a modificar. Si el usuario existe la capa de negocio realiza una nueva consulta a la base de datos, solicitando toda la información almacenada en la tabla de restaurantes de la base de datos del registro con el id recibido en modo oculto.

Se recibe la respuesta de la consulta con todos los datos de los campos del registro encontrado en la base de datos con el id del restaurante que se quiere editar. Se rellena los campos del formulario con la información recibida.

Una vez el usuario pulsa el botón se ejecuta la acción del formulario, la cual recarga la pagina enviando en modo oculto todos los campos del formulario. Al recargar la pagina se comprueba la autenticidad del usuario mediante la consulta mencionada anteriormente, si el usuario es un usuario valido se pasa a comprobar si se ha recibido el formulario para la modificación de los campos del restaurante. En este caso si se ha recibido, se realiza el mismo proceso descrito en la creación de un restaurante, se comprueban los campos obligatorios, los horarios y la foto. La diferencia esta al insertar los datos en la tabla, al terminar la comprobación de los campos del formulario recibido se pasa a modificar los campos del registro en la tabla restaurantes de la base de datos con el mismo id recibido.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {
 Consulta Validar (formulario (“usuario”), formulario
 (“contraseña”));

SI es valido {

 SI se ha recibido form_editar /*el formulario para crear un
 nuevo restaurante*/ {

 SI (longitud (form_editar (“entrada_indice”)>0 Y
 longitud (form_editar (“entrada_indice”) <=200 Y
 Validar (form_editar (“entrada_indice”))) {
 Datos_editar [(“entrada_indice”)] =
 form_editar (“entrada_indice”);
 } SI NO {
 Error = verdadero;
 }
 }

 SI (longitud (form_editar (“dirección”)>0 Y longitud
 (form_editar (“dirección”) <=200 Y Validar
 (form_editar (“dirección”))) {
 Datos_editar [(“dirección”)] = form_editar
 (“dirección”);
 } SI NO {
 Error = verdadero;
 }
}

 SI (longitud (form_editar (“ciudad”)>0 Y longitud
 (form_editar (“ciudad”) <=150 Y Validar (form_editar
 (“ciudad”))) {
 Datos_editar [(“ciudad”)] = form_editar
 (“ciudad”);
 } SI NO {
 Error = verdadero;
 }
}

 SI (longitud (form_editar (“telefono”)>0 Y longitud
 (form_editar (“telefono”) <=12 Y Validar (form_editar
 (“telefono”))) {

```
Datos_editar[“telefono”] = form_editar
(“telefono”);
} SI NO {
    Error = verdadero;
}

DESDE x = 1 HASTA x <= form_editar
(“total_mañana”) {
    SI x != 1 {Horario_mañana = Horario_mañana +
“$”}
        SI (Valido (form_editar
(“horario_mañana”+x”+”_dia”))) Y form_editar
(“horario_mañana”+x”+”_dia”) <=150 Y (form_editar
(“horario_mañana”+x”+”_hora”))) Y form_editar
(“horario_mañana”+x”+”_hora”) <=150)) {
            Horario_mañana = Horario_mañana +
form_editar (”horario_mañana”+x”+”horario_dia”) + “#” +
form_editar (”horario_mañana”+x”+”horario_hora”);
        }
    }

DESDE x = 1 HASTA x <= form_editar (”total_tarde”)
{
    SI x != 1 {Horario_tarde = Horario_tarde + “$”}
        SI (Valido (form_editar (”horario_
tarde”+x”+”_dia”))) Y form_editar (”horario_
tarde”+x”+”_dia”) <=150 Y (form_editar (”horario_
tarde”+x”+”_hora”))) Y form_editar (”horario_
tarde”+x”+”_hora”) <=150)) {
            Horario_tarde = Horario_tarde +
form_editar (”horario_tarde”+x”+”horario_dia”) + “#” +
form_editar (”horario_tarde”+x”+”horario_hora”);
        }
    }

SI (longitud (form_editar (”primero”))>0 Y longitud
(form_editar (”primero”) <=3 Y Validar (form_editar
(”primero”))) {
        Datos_editar[”primero”] = form_editar
(”primero”);
    } SI NO {
        Error = verdadero;
    }
```

```
    }

    SI (form_editar ("foto")) {
        Datos_editar["foto"]      =      form_editar
        ("foto");
    } SI NO {
        Datos_editar["foto"] = foto_porvisonal;}

    SI (form_editar ("foto")) {
        Prefijo = generar_tres_numero_con_MD5;
        Direccion_foto = prefijo + "_" + form_editar
        ("foto").nombre;

        SI      subir_foto      (form_editar      ("foto"),
        direccion_foto){
            Resultado = Crear_thumb (Direccion_foto);
            SI (resultado = 1) {
                Mensaje = "La imagen" +
                form_editar ("foto").nombre + "a sido creda
                correctamente";
            } SI NO {
                Mensaje = "La imagen es incorrecta";
                Error_foto = verdad;
            }
        } SI NO {
            Mensaje = "Ha ocurrido un error con el servidor";
            Error_foto = verdad;
        }
    }

    SI (NO error Y NO error_foto) {
        Actualizar el registró de la tabla "tb_restaurantes" con
        los datos procesados, donde id = form_editar ("id");
        Mensaje = "Las modificaciones se han guardado
        correctamente."
    }

    Array datos_restaurante = Consulta recuperar datos del
    restaurante con id=form_editar ("id") de la tabla "tb_noticias";

    Horario_mañana =      separar      ("$",datos_restaurante
    ["horario_mañana"]);
```

```
Horario_tarde = separar ("$",datos_restaurante ["horario_
tarde"]);

DESDE j =1 HASTA j<= total (Horario_mañana) {
    Horario_m_aux [j] = separar ("#", Horario_mañana[j]);
}
DESDE j =1 HASTA j<= total (Horario_tarde) {
    Horario_t_aux [j] = separar ("#", Horario_tarde [j]);
}

SI (mensaje) {imprimir mensaje};
Consulta recuperar los datos del restaurante con el id =
form_editar ["id"];
Imprimir variables en el formulario de edición del restaurante;

} SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

4.4.3.4- Gestor Noticias

Mismo proceso de inicio que en las demás paginas del gestor. Se comprueba que los datos de autenticación del usuario son correctos mediante una consulta a la base de datos. Si no se han recibido dichos datos, o si son erróneos, se redirige a la pagina web de inicio del gestor.

Una vez se comprueba que el usuario es correcto, se pasa a comprobar si se ha recibido el formulario para la eliminación de una noticia, al no recibirse se pasa a comprobar si se ha recibido el formulario para la modificación de la noticia destacada, como no es así se continua sin realizar mas acciones.

Se realizan una consulta a la base de datos para recoger todos los registros de la tabla “tb_noticias”. Se obtiene la respuesta de la consulta y se pasa a generar el código HTML de la tabla empleada como índice de las noticias, en donde cada noticia es representada por su titular. La tabla además contiene una columna de acciones, con las acciones “editar” y “borrar”. Cada acción es un enlace que contiene una sentencia en JavaScript que llama una función JavaScript, encargada de realizar “submit” en los formularios de edición de noticias y eliminación de las mismas.

La tabla tiene una columna de noticias destacadas, que se va generando según se generan el índice de restaures y sus acciones. Se comprueba si la noticia para la que se va a escribir el código HTML en el índice es la noticia destacada, si lo es se pone como marcado en esta columna.

Si el usuario pulsa el enlace “borrar” de la tabla para eliminar alguna noticia, se realiza un “submit” del formulario de borrado de noticias. Dicho formulario posee 3 campos, el e-mail del usuario, la contraseña del usuario y el id del noticia a modificar. El campo id del formulario para eliminar noticias cambio de valor al pulsar el enlace borrar, el cual contiene el id de la noticia que se encuentra en la misma fila de la tabla. Al realizar “submit” en el formulario para eliminar noticias se carga la misma página en la que se encuentra el usuario, pero esta vez se recibe en modo oculto el formulario para la eliminación de una noticia.

Se vuelve a comprobar que los datos de acceso del usuario son correctos, si no se redirige al usuario a la página de inicio del gestor. Si son correctos se comprueba que se ha recibido el formulario para la eliminación de una noticia y se pasa a enviar, mediante PHP, la solicitud de eliminación del registro con el id recibido en la tabla de noticias de la base de datos y se elimina. Se comprueba si se ha recibido el formulario para modificar la noticia destacada, como no es así se continua con la ejecución normal de la pagina. Se vuelve a consultar todos los registros de la tabla noticias, en este caso sin la noticia eliminada.

Si el usuario quiere modificar la noticia destacada ha de pulsar el botón en la columna “Noticia destacada” de la tabla que corresponda con la noticia que se a cambiar a noticia destacada.

Una vez se pulsa el botón aparece un botón para guardar el resultado. Dicho botón es un enlace con una sentencia JavaScript que llama a una función JavaScript, la cual modifica el valor id del formulario creado para modificar la noticia destacada y realiza un “submit” del formulario, recargando la pagina.

Al recargar la página se vuelve a comprobar que los datos de acceso del usuario son correctos y se ha recibido el formulario de eliminación de una noticia. El usuario es correcto y no se recibido el formulario para la eliminación de una noticia, se comprueba que si se ha recibido el formulario para la modificación de la noticia destacada con los datos del usuario y el id de la próxima noticia destacada. Se pasa a modificar el campo “primero” a cero, el cual representa la noticia destacada en la base de datos, en todos los registros de la tabla noticias de la base de datos. Cuando todas las noticias tienen el campo “primero” igual a cero, se realiza otra sentencia PHP, pero esta vez se solicita a la base de datos que cambie el valor del campo “primero” a uno en el registro en la tabla de noticias con el mismo id que el id recibido.

Ya actualizada la noticia destacada se hace la consulta ala base de datos para cargar los datos de todos los registros de la de noticias y generar de nuevo la tabla con el índice de noticias, acciones y noticia destacada, viéndose la nueva noticia destacada.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {

 Validar (formulario (“usuario”), formulario (“contraseña”));

 SI es valido {

 SI se ha recibido el formulario para borrar {

 Borrar_noticia (id recibido);

 }

 SI se ha recibido el formulario para cambiar noticia destacada

 {

 Actualizar todos los registros de la tabla “tb_noticias” a no destacados;

 Actualizar la variable “destacado” a 1 del registro de la tabla “tb_noticias” con el id = “id del formulario recibido “;

 }

 Array datos_noticias = Consulta recuperar datos de todas las noticias de la tabla “tb_noticias”;

 MIENTRAS existan datos en el Array datos_noticias {

 Imprimir datos_noticias (“entrada_indice”);

 Imprimir enlace que llama a la función editar (datos_noticias (“id”));

 Imprimir enlace que llama a la función borrar (datos_noticias (“id”));

 SI (datos_noticias[destacado]=1){

 Imprimir input radio con checked = “checked, acción al clicar = invocar función noticia_primer (datos_noticias[“id”]) y valor = datos_noticias[“id”];

 } SI NO {

 Imprimir input radio con acción al clicar = invocar función noticia_primer (datos_noticias[“id”]) y valor = datos_noticias[“id”];

 }

 }

 } SI NO {redirigir a la página de inicio del gestor;}

} SI NO {redirigir a la página de inicio del gestor;}

4.4.3.5 - Nueva Noticia

Al acceder a esta página se comprueba si se han recibido los datos de acceso del usuario y si son correctos mediante la consulta a la base de datos mencionada en la demás secciones. En caso negativo de cualquier de los dos casos se redirige a la pagina inicial del gestor.

Seguidamente se comprueba si se ha recibido el formulario para la creación de un nueva noticia, como no se ha recibido no se realiza ninguna acción mas en la capa de negocio.

Una vez el usuario rellena correctamente el formulario para la creación de una nueva noticia, pulsa el botón de enviar y se comprueban todos los campos del formulario sin error, se ejecuta la sentencia “submit” del formulario de creación de nuevas noticias, en donde aparte de contener los datos necesarios para la creación de una nueva noticia se dispone de los datos de acceso del usuario, y se vuelve a cargar la pagina actual pero enviando el formulario para la creación de noticias.

Al recargar la pagina se vuelve a comprobar que se han recibido los datos de acceso del usuario y además si son correctos. Si todo va bien se comprueba que se ha recibido el formulario para la creación de una nueva noticia. Como se ha recibido se comprueba campo a campo si contiene información, dando error en caso de ser un capo obligatorio. En caso de estar rellenos se comprueba que no contenga sentencias SQL, para evitar SQL Injection. Si tuviera sentencias SQL no se realiza ninguna acción. Si no ha habido problemas se envía a la capa de datos la solicitud de creación de una nueva noticia en la base de datos, creándola sin problemas. En caso de que la noticia contenga una foto se realiza el mismo proceso que el descrito al crear un restaurante, incluyendo la creación de una imagen en caso de tener un tamaño inadecuado.

Una vez se creado correctamente la noticia, se notifica al usuario a través de un mensaje que parece debajo del titulo de la sección que la noticia ha sido creada correctamente.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {
 Consulta Validar (formulario (“usuario”), formulario
 (“contraseña”));

SI es valido {

 SI se ha recibido form_nuevo /*el formulario para crear una
 nueva noticia*/ {

 SI (longitud (form_nuevo (“entrada_indice”)>0 Y
 longitud (form_nuevo (“entrada_indice”) <=200 Y
 Validar (form_nuevo (“entrada_indice”))) {
 Datos_nuevo[(“entrada_indice”) =
 form_nuevo (“entrada_indice”);
 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“fuente”)>0 Y longitud
 (form_nuevo (“fuente”) <=200 Y Validar (form_nuevo
 (“fuente”))) {
 Datos_nuevo[(“fuente”) = form_nuevo
 (“fuente”);
 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“fecha”)>0 Y longitud
 (form_nuevo (“fecha”) <=10 Y Validar (form_nuevo
 (“fecha”))) {
 Datos_nuevo[(“fecha”) = form_nuevo
 (“fecha”);
 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“contenido”)>0 Y Validar
 (form_nuevo (“contenido”))) {
 Datos_nuevo[(“contenido”) = form_nuevo

```
        ("contenido");
    } SI NO {
        Error = verdadero;
    }

    SI (form_nuevo ("foto")) {
        Prefijo = generar_tres_numero_con_MD5;
        Direccion_foto = prefijo + "_" + form_nuevo
        ("foto").nombre;

    SI (longitud (form_nuevo ("destacada"))>0) {
        Datos_nuevo["destacada"] = form_nuevo
        ("destacada");
    } SI NO {
        Error = verdadero;
    }

    SI (longitud (form_nuevo ("link"))>0 Y longitud
    (form_nuevo ("link")) <=500 Y Validar (form_nuevo
    ("link"))) {
        Datos_nuevo["link"] = form_nuevo
        ("link");
    } SI NO {
        Error = verdadero;
    }

    SI (form_nuevo ("foto")) {
        Datos_nuevo["foto"] = form_nuevo
        ("foto");
    } SI NO {
        Datos_nuevo["foto"] = foto_porvisonal;}

    SI subir_foto (form_nuevo ("foto"),
    direccion_foto){
        Resultado = Crear_thumb (Direccion_foto);
        SI (resultado = 1) {
            Mensaje = "La imagen" +
            form_nuevo ("foto").nombre + "a sido creada
            correctamente";
        } SI NO {
            Mensaje = "La imagen es incorrecta";
```

```
                Error_foto = verdad;
            }
        } SI NO {
            Mensaje = "Ha ocurrido un error con el servidor";
            Error_foto = verdad;
        }

    SI (NO error Y NO error_foto) {
        Crear registró en la tabla "tb_noticias" con los datos
        procesados;
        Mensaje = "La noticia ha sido creada correctamente";
    }

    SI (mensaje) {imprimir mensaje};

    } SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

4.4.3.6 - Editar Noticia

Cuando un usuario abre la pagina de edición de noticias la capa de negocio solicita los datos de acceso del usuario que accede a la pagina, si no se dispone de ellos se redirige a la pagina de inicio del gestor. Si se poseen los datos de acceso del usuario se comprueba mediante una consulta a la base de datos que los datos del usuario corresponden con un los datos de un registro de la tabla usuarios, y que son correctos. En caso negativo se procede a redirigir al usuario la página de inicio del gestor.

Si no ha habido error y el usuario es correcto la capa de negocio comprueba si se ha recibido el formulario para la edición de los registros de noticias. Al ser el primer acceso a la página no se ha podido recibir ningún formulario para la modificación de los datos de un registro de la tabla noticias de la base de datos y se continúa con la ejecución normal de la página.

Se comprueba si se generado el mensaje que confirma que la edición de la noticia a sido correcta, el cual no se ha generado ya que no se ha modificado ningún registro de la base de datos.

Se hace una consulta ala base de datos pidiendo los datos del registro de la tabla de noticias en donde el id recibido es igual al id del registro buscado. Se recibe la respuesta y a continuación se rellena el formulario con los datos obtenidos, cumplimentando de esta manera los datos del formulario para la edición de noticias.

Cuando el usuario modifica la noticia y envía el formulario de edición de noticias, el cual contiene los datos s de acceso del usuario, se recarga la página. Se comprueba si se han recibido los datos de acceso del usuario y si son correctos, si no hay problema se comprueba si se ha recibido el formulario para la edición de noticias, en este caso si se recibe y se comprueban los campos, comprobando que los campos obligatorios no estén vacíos y que todos los campos no vacíos sean validos, es decir no contengan sentencias SQL. Si todo es correcto, mediante un sentencia PHP, se envían los datos de la notica a modificar y se solicita a la capa de datos que modifique los campos de la notica con el mismo id que el enviado en los datos de la noticias, suplantando el resto de datos de los campos por los datos recibidos para cada campo del registro con ese id.

Se genera el mensaje de confirmación de modificación correcta, a continuación la capa de negocio, comprueba si se generó dicho mensaje y lo muestra. De esta manera el usuario sabe con certeza que la noticia ha sido modificada correctamente.

Pseudocódigo:

Cargar librerías;

```
SI se ha recibido el formulario para logarse {  
    Consulta    Validar    (formulario    ("usuario"),    formulario  
    ("contraseña"));
```

```
    SI es valido {
```

```
        SI se ha recibido form_editar /*el formulario para crear una  
        nueva noticia*/ {
```

```
            SI (longitud (form_editar ("entrada_indice"))>0 Y  
            longitud (form_editar ("entrada_indice")) <=200 Y  
            Validar (form_editar ("entrada_indice"))) {  
                Datos_editar[("entrada_indice")] =  
                form_editar ("entrada_indice");  
            } SI NO {  
                Error = verdadero;  
            }  
        }
```

```
        SI (longitud (form_editar ("fuente"))>0 Y longitud  
        (form_editar ("fuente")) <=200 Y Validar (form_editar  
        ("fuente"))) {  
            Datos_editar[("fuente")] = form_editar  
            ("fuente");  
        } SI NO {  
            Error = verdadero;  
        }  
    }
```

```
    SI (longitud (form_editar ("fecha"))>0 Y longitud  
    (form_editar ("fecha")) <=10 Y Validar (form_editar  
    ("fecha"))) {  
        Datos_editar[("fecha")] = form_editar  
        ("fecha");  
    } SI NO {  
        Error = verdadero;  
    }  
}
```

```
SI (longitud (form_editar ("contenido"))>0 Y Validar
(form_editar ("contenido"))) {
    Datos_editar["contenido"] = form_editar
    ("contenido");
} SI NO {
    Error = verdadero;
}
```

```
SI (longitud (form_editar ("destacada"))>0) {
    Datos_editar ["destacada"] = form_editar
    ("destacada");
} SI NO {
    Error = verdadero;
}
```

```
SI (longitud (form_editar ("link"))>0 Y longitud
(form_editar ("link")) <=500 Y Validar form_editar
("link")) {
    Datos_editar["link"] = form_editar ("link");
} SI NO {
    Error = verdadero;
}
```

```
SI (form_editar ("foto")) {
    Datos_editar["foto"] = form_editar
    ("foto");
} SI NO {
    Datos_editar["foto"] = foto_porvisonal;}
```

```
SI (form_editar ("foto")) {
    Prefijo = generar_tres_numero_con_MD5;
    Direccion_foto = prefijo + "_" + form_editar
    ("foto").nombre;
    SI subir_foto (form_editar ("foto"),
    direccion_foto){
        Resultado = Crear_thumb (Direccion_foto);
        SI (resultado = 1) {
            Mensaje = "La imagen" +
            form_editar ("foto").nombre + "a sido creada"
```

```
correctamente";
        } SI NO {
            Mensaje = "La imagen es incorrecta";
            Error_foto = verdad;
        }
    } SI NO {
        Mensaje = "Ha ocurrido un error con el servidor";
        Error_foto = verdad;
    }

    SI (NO error Y NO error_foto) {
        Actualizar el registró de la tabla "tb_noticias" con los
        datos procesados, donde id = form_editar ["id"];
        Mensaje = "Las modificación se han guardado
        correctamente.";
    }

    SI (mensaje) {imprimir mensaje};
    Consulta recuperar los datos de la tabal "tb_noticias" con el id
    = form_editar ["id"];
    Imprimir variables en el formulario de edición de la noticia;

    } SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

4.4.3.7 – Gestor Banner

Al acceder a la página de gestión del banner la capa de negocio requiere los datos de acceso del usuario, si no se dispone de ellos se redirige a la página de inicio del gestor. Si se dispone de los datos de acceso del usuario la capa de negocio hace una consulta a la capa de datos solicitando los datos del usuario de la tabla de usuario que corresponda con los datos del usuario, si existe se continúa con la ejecución de la página, si no se redirige al usuario a la página principal del gestor.

La capa de negocio verifica si a través del método “POST” se ha recibido el formulario para la eliminación de un registro de la tabla “tb_banner”. Al no recibir el formulario se continúa con la ejecución normal de la página.

La capa de negocio revisa si se ha enviado el formulario para modificar el tiempo que transcurre del pase de una diapositiva a otra en el banner, al no recibirse se continúa con la ejecución de la página.

Una vez finalizadas las comprobaciones del comienzo de la página la capa de negocio se pasa a generar el código de la tabla que contiene el índice de dispositivas del banner almacenadas en el sistema, para ello crea la primera fila de la tabla, en donde se indica la función de cada columna. A continuación se realiza una consulta a la base pidiendo los datos de todos los registros de la tabla banner. Una vez se recibe la respuesta por parte de la capa de datos se pasa a continuar generando la tabla, en donde cada fila de la tabla representa una diapositiva del banner. Para diferenciar las diapositivas en el índice se emplea el id de cada diapositiva como referencia en el índice.

La tabla además contiene los enlaces para editar y borrar los registros de las diapositivas del banner, los cuales contienen sentencias JavaScript que invocan funciones. En el caso del enlace editar, la función invocada cambia el valor de la variable que almacena el id del registro a editar y ejecuta la sentencia “submit” sobre el formulario de edición para editar el registro de una diapositiva.

En el caso del enlace creado para borrar registros de diapositivas en la base de datos se modifica el valor de la variable que contiene el id del registro de la tabla “tb_banner” que representa la diapositiva a eliminar y realiza un “submit” del formulario creado para eliminar un registro de la tabla “tb_banner”.

Para finalizar la creación de la tabla se genera un enlace con el nombre “Nueva diapositiva”, dicho enlace posee una sentencia JavaScript, la cual llama a una función JavaScript que ejecuta la “submit” sobre el formulario para creación de diapositivas.

Todos los formularios anteriormente mencionados redirige al usuario a las paginas de destino, y envían los datos de acceso del usuario, y en caso de los formularios para la eliminación y edición de registros de diapositivas envían además el id del registro a eliminar o modificar.

Una vez creada la tabla la capa de negocio hace una consulta a la capa de datos solicitando la información del registro de la tabla “tb_tiempos” en donde la variable “banner” sea igual a “banner_1”. Una vez se recibe la respuesta se genera el pequeño formulario al final de la pagina en donde se muestra el tiempo que transcurre en el paso de diapositivas.

Si el usuario elimina una diapositiva el formulario es redirigido a la misma página, pero esta vez enviando el formulario para eliminación de un registro de una diapositiva. Al recargase la pagina se vuelven a verificar la autenticidad de los datos de acceso del usuario. Se verifica que se ha recibido el formulario de eliminación de diapositivas, con el id del registro a eliminar. Mediante PHP se indica a la capa de negocio que registro eliminar en la tabla “tb_banner” de la base de datos, el registro con el id recibido. Una vez eliminado el registro se comprueba si se ha recibido el formulario para la modificación del tiempo que transcurre en el paso de diapositivas, como no se ha recibido se continua con la ejecución de la pagina, en donde el usuario puede comprobar que la diapositiva eliminado no aparece ya en el índice de la tabla.

Si el usuario modifica el tiempo que pasa en el paso de diapositivas del banner, se recarga la página enviando el formulario creado para modificar dicho tiempo. Al volver a cargar la página se comprueba la autenticidad de los datos de acceso del usuario, se comprueba que no se ha recibido el formulario para la eliminación de diapositivas del banner, y se verifica que se ha recibido el formulario para modificar en el tiempo que transcurre en el paso de una diapositiva a otra. Se solicita a la capa de datos que modifique el tiempo del registro de la tabla “tb_tiempos” en donde el nombre del banner sea “banner_1”. Una vez modificado se genera la tabla y el tiempo de paso de diapositivas.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {

 Validar (formulario (“usuario”), formulario (“contraseña”));

 SI es valido {

 SI se ha recibido el formulario para borrar {

 Borrar_diapositiva (id recibido);

 }

 SI se ha recibido el formulario para cambiar el tiempo {

 Actualizar tiempo = frm_tiempo [“tiempo”] del registro
de la tabla “tb_tiempos” donde nombre = “banner_1”;

 }

 Tiempo = consulta recuperar tiempo de la tabla “tb_tiempos”
donde nombre = “banner_1”;

 Array datos_banner = Consulta recuperar los datos de todos
los registros de la tabla “tb_benner”;

 MIENTRAS existan datos en el Array datos_banner {

 Imprimir datos_banner (“id”);

 Imprimir enlace que llama a la función editar
(datos_banner (“id”));

 Imprimir enlace que llama a la función borrar
(datos_banner (“id”));

 }

 Imprimir, en el input de texto que almacena el tiempo que
transcurre entre el paso de dispositivas del banner, el valor de
la variable Tiempo;

 } SI NO {redirigir a la página de inicio del gestor;}

} SI NO {redirigir a la página de inicio del gestor;}

4.4.3.8 – Nueva dispositiva

Al entrar en la página la capa de negocio realiza el mismo proceso de verificación de los datos de acceso del usuario, comprobando que se han recibido y que son correctos mediante la consulta anteriormente mencionada a la capa de datos. Si alguna comprobación falla se redirige al usuario a la página de inicio del gestor.

A continuación la capa de negocio examina si se ha recibido el formulario la creación de un nuevo registro en la tabla “tb_banner” de la base de datos. Como aun no se ha recibido, por ser el primer acceso a la página, se continúa con la ejecución de la página.

Para crear una nueva diapositiva en el banner el usuario a de rellenar el formulario para la creación de la nueva diapositiva, si todo es correcto se recarga la página actual, pero esta vez se envía el formulario para la creación de una diapositiva, con los datos de acceso del usuario. Al recargarse la pagina se vuelve a verificar la autenticidad del usuario, si no hay problemas se comprueba que si se a recibido el formulario de creación de una nuevo diapositiva, por lo tanto se envía a la capa de datos la solicitud de crear un nuevo registro en le tabla “tb_banner” y los datos de todos los campos que lo forman.

Pseudocódigo:

Cargar librerías;

SI se a recibido el formulario para logarse {

 Consulta Validar (formulario (“usuario”), formulario
 (“contraseña”));

 SI es valido {

 SI se ha recibido form_nuevo /*el formulario para crear una
 nueva noticia*/ {

 SI (longitud (form_nuevo (“contenido”)>0 Y Validar
 (form_nuevo (“contenido”))) {

 Datos_nuevo[(“contenido”) = form_nuevo
 (“contenido”);

 } SI NO {

 Error = verdadero;

 }

 SI (longitud (form_nuevo (“posición_contenido”)) {

 Datos_nuevo[(“posición_contenido”) =

```
        form_nuevo ("posición_contenido");
    } SI NO {
        Error = verdadero;
    }

    SI (form_nuevo ("foto")) {
        Prefijo = generar_tres_numero_con_MD5;
        Direccion_foto = prefijo + "_" + form_nuevo
("foto").nombre;

        SI      subir_foto      (form_nuevo      ("foto"),
direccion_foto){
            Resultado = Crear_thumb (Direccion_foto);
            SI (resultado = 1) {
                Mensaje = "La imagen" +
form_nuevo      ("foto").nombre + "a sido creada
correctamente";
            } SI NO {
                Mensaje = "La imagen es incorrecta";
                Error_foto = verdad;
            }
        } SI NO {
            Mensaje = "Ha ocurrido un error con el servidor";
            Error_foto = verdad;
        }

        SI (NO error Y NO error_foto) {
            Crear registró en la tabla "tb_banner" con los datos
procesados;
            Mensaje = "La dispositiva ha sido creada
correctamente";
        }

        SI (mensaje) {imprimir mensaje};

    } SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

4.4.3.9 – Editar diapositiva

La capa de negocio comprueba la autenticidad de los datos con los que el usuario ha accedido al gestor, primero se verifica que se han recibido dichos datos y después, mediante una consulta a la base de datos, se comprueba que los datos corresponden con los datos de un usuario dado de alta en el sistema. En caso de que alguna comprobación falle se envía al usuario a la página de inicio del gestor.

Se pasa a comprobar si se ha recibido el formulario para la edición de los datos de un banner, en este caso aun no se han recibido y se no se realiza ninguna acción en este punto. También se comprueba si existe una variable que se crea al modificar satisfactoriamente una diapositiva, como no se ha creado no se realiza ninguna acción en este punto.

Seguidamente la capa de negocio solicita a la capa de datos toda la información almacenada en la base de datos del registro de la tabla “tb_banner” con el mismo id que el id recibido al cargar la pagina. Se obtiene la respuesta por parte de la capa de datos con todos los campos del registro solicitado. La capa de negocio rellena el formulario donde se encuentran los campos que forman los registros de las diapositivas con los datos recibidos.

En cuanto el usuario pulse el botón de enviar, la capa de presentación comprueba mediante JavaScript que los datos sean correctos, si es así se pasa a recargar la pagina actual enviando el formulario con los datos modificados de la diapositiva del banner, mas los datos de acceso del usuario y el id del banner que se esta modificando. Al recargar la página se vuelve a verificar la autenticidad de los datos de acceso del usuario. Si no hay error la capa de negocio verifica que se ha recibido el formulario para la edición de la diapositiva. En este caso se comprueba uno a uno que los campos sean correctos, incluyendo el campo foto, la comprobación de este campo se ha explicado anteriormente. En caso de que todos los campos sean correctos y no haya fallo se pide a la capa de datos que modifique los campos del registro de la tabla “tb_banner” con el mismo id que el enviado. Los datos de los campos que se modifican son enviados con la solicitud de modificación de los datos de la diapositiva.

Una vez modificados los datos de la diapositiva se genera un mensaje en el que se indica que los datos han sido modificados correctamente.

Se verifica que la variable mensaje existe, y se muestra.

La capa de negocio vuelve a consultar a la capa de datos los datos de la diapositiva actual y los carga en el formulario, mostrándoselos al usuario.

Pseudocódigo:

Cargar librerías;

SI se ha recibido el formulario para logarse {
 Consulta Validar (formulario (“usuario”), formulario
 (“contraseña”));

 SI es valido {
 SI se ha recibido form_editar /*el formulario para crear una
 nueva noticia*/ {

 SI (longitud (form_nuevo (“contenido”)>0 Y Validar
 (form_nuevo (“contenido”))) {
 Datos_nuevo[(“contenido”] = form_nuevo
 (“contenido”);

 } SI NO {
 Error = verdadero;
 }

 SI (longitud (form_nuevo (“posición_contenido”)) {
 Datos_nuevo[(“posición_contenido”] =
 form_nuevo (“posición_contenido”);
 } SI NO {
 Error = verdadero;
 }

 SI (form_editar (“foto”)) {
 Prefijo = generar_tres_numero_con_MD5;
 Direccion_foto = prefijo +”_” + form_editar
 (“foto”).nombre;

 SI subir_foto (form_editar (“foto”),
 direccion_foto){
 Resultado = Crear_thumb (Direccion_foto);
 SI (resultado = 1) {
 Mensaje = “La imagen” +
 form_editar (“foto”).nombre + “a sido creada
 correctamente”;

```
        } SI NO {
            Mensaje = "La imagen es incorrecta";
            Error_foto = verdad;
        }
    } SI NO {
        Mensaje = "Ha ocurrido un error con el servidor";
        Error_foto = verdad;
    }

    SI (NO error Y NO error_foto) {
        Actualizar el registró de la tabla "tb_banner" con los
        datos procesados, donde id = form_editar ("id");
        Mensaje = "Las modificación se han guardado
        correctamente.";
    }

    SI (mensaje) {imprimir mensaje};
    Consulta recuperar los datos de la con el id = form_editar
    ["id"];
    Imprimir variables en el formulario de edición del banner;

    } SI NO {redirigir a la página de inicio del gestor;}
} SI NO {redirigir a la página de inicio del gestor;}
```

4.5.- Capa de Datos

Es la capa donde residen los datos y es la encargada de acceder a los mismos. Está formada por la base de datos, donde se almacenan los datos de los que se nutren las secciones gestionables de la pagina web. La capa de datos realiza todo el almacenamiento de datos para las secciones gestionables.

Es importante resaltar que la capa de datos solo se comunica con la capa de negocio. Reciben solicitudes de almacenamiento, modificación, eliminación o recuperación de información desde la capa de negocio.

La base de datos contiene sentencias de control para los campos de las tablas que la forman. Estas sentencias de control impiden almacenar formatos de datos que no correspondan con los formatos establecidos para los campos de las tablas.

Para que una variable sea almacenada en un campo de una tabla de la base datos se requiere que sea de mismo tipo, es decir nunca se podrá almacenar un numero de tipo Integer en un campo de tipo Char de la base de datos. De esta manera la capa de datos mantiene su estructura e impide posible errores, manteniendo su integridad intacta.



5.- BATERIA DE PRUEBAS

5.1.- Introducción

Una vez creada la aplicación web se realiza la batería de pruebas. En esta sección se muestra un resumen la batería de pruebas realizadas sobre la aplicación web una vez finalizada su implementación.

A lo largo de la implementación de la aplicación Web se han realizando pruebas según se ha ido implementando las distintas secciones de la página Web que componen la aplicación Web.

También se han probando los procedimientos creados en la base de datos utilizados en la aplicación Web. Por lo que se han corrigiendo errores desde el principio de la implementación, resultando la tarea de probar la aplicación una tarea fácil y poco costosa, a lo largo de la implementación.

A pesar de haber hecho pruebas durante la implementación se ha creado esta batería de pruebas para asegurar el correcto funcionamiento y la integridad de la seguridad de la aplicación web.

La batería de pruebas se ha efectuado en un servidor local, corrigiendo errores sintácticos, errores de navegación, etc.

En general, tal y como se especifica a continuación, los resultados han sido favorables y las pruebas han resultado claramente satisfactorias.

5.2.- Visualización

Para que la página web funcione y sea visitada es necesario que sea visible en los navegadores mas utilizados por los usuarios de internet, así como en diversas resoluciones, Smart fones y tabletas digitales con conexión a internet.

En los requisitos se especifica que el tamaño de resolución mínimo sobre el que se puede visualizar la página web corresponde a 1024x768, siendo la resolución mínima para la batería de pruebas.

Exploradores de internet	1024x768	1280x720	1280 x 960	1600x900	1920x1080	2560x1600
Mozilla Firefox	✓	✓	✓	✓	✓	✓
Google Chrome	✓	✓	✓	✓	✓	✓
Internet Explorer	✓	✓	✓	✓	✓	✓
Safari	✓	✓	✓	✓	✓	✓
Opera	✓	✓	✓	✓	✓	✓
Safari en iPhone	✓					
Safari en iPad	✓					
Android HTC Desire HD	✓					
Android Samsung Galaxy Ace	✓					

Tabla 53 Pruebas visualización

5.3.- Funcionamiento

Se realizan las pruebas sobre el gestor, introduciendo, modificando y eliminando el contenido de cada sección, y después se comprueba sobre la página web. De esta manera se consigue realizar el test sobre la introducción, modificación, eliminación y visualización de los datos de las diferentes secciones.

Primero se comprueba el proceso login, en donde se incluyen funciones para evitar SQL injection y Cross-site Scripting, y mensajes indicando que los datos son incorrectos o no cumplen con las condiciones para serlo en caso de ser incorrectos.

Pruebas	E-mail	Contraseña
Mensaje de error si esta vacío	✓	✓
Mensaje de error si no tiene formato estándar	✓	✗
Anti SQL-Injection	✓	✓
Anti Cross-site Scripting	✓	✓

Tabla 54 Pruebas login

No existe un mensaje de error en caso de que el usuario no introduzca una contraseña estándar porque no hay un tipo de contraseña estándar, a diferencia del e-mail, cuyo formato es; “caractares”@”caranteres”.”caractares”.

Se pasa a introducir datos erróneos en el gestor. Se espera que el gestor detecte dichos datos erróneos y genere un mensaje de error indicando dicho error:

Secciones	Datos obligatorios vacíos	Datos de tipo incorrecto	Foto a subir no es un tipo de imagen
Sección restaurantes	✓	✓	✓
Sección noticias	✓	✓	✓
Sección banner	✓	✓	✓

Tabla 55 Pruebas en el gestor I

Para comprobar que el gestor y la base de datos funcionan correctamente se van a insertar, modificar y eliminar datos en las secciones del gestor y después se comprobará que dichos cambios se efectúan en la página web principal del proyecto. Además se van a realizar pruebas sobre los formularios del gestor y de la página web para confirmar su seguridad, imposibilitando los ataques mediante SQL Injection y Cross-site Scripting.

	Insertión	Modificación	Eliminación	Seguridad ante SQL Injection	Seguridad ante Cross-site Scripting
Sección restaurantes	✓	✓	✓	✓	✓
Sección noticias	✓	✓	✓	✓	✓
Sección banner	✓	✓	✓	✓	✓
Visualización de los cambios en la página web	✓	✓	✓	✓	✓

Tabla 56 Pruebas en el gestor II

Los índices de la pagina principal tiene un espacio asignado, de un tamaño siempre constante, por motivos de espacio, interesa que el contenido de la sección no sea colapsado por la visualización de grandes índices. Cuando un índice supera el tamaño del espacio asignado aparecen dos enlaces, “más” y “menos”, para poder recorrer todo el índice sin aumentar el espacio que le ha sido asignado. Estos índices residen en la secciones restaurantes y noticias, en cuanto al banner no preciosa de índice por ser un deslizador de diapositivas.

También se prueban las rutas de las imágenes, que los ficheros adjuntos se almacenen bien en la base de datos y en el servidor, y que los correos electrónicos de la sección contacto sean mandados correctamente al correo electrónico de la franquicia con el formato dado.



6.- CONCLUSIONES

6.1.- Conclusiones

Con el proyecto de fin de carrera ya finalizado se puede realizar un análisis en profundidad de lo logrado a lo largo de todas las fases de creación de la aplicación web.

Analizando en profundidad el proyecto de fin de carrera se puede decir que se ha logrado desarrollar un portal web comercial que ha cubierto todas las necesidades y ha cumplido todos los objetivos impuestos al comienzo de su desarrollo, con un coste y tiempos de desarrollo más que aceptables.

De cara a los usuarios, el portal web se ha desarrollado para que pueda ser utilizado fácilmente y de manera intuitiva por los usuarios, empleando una navegación horizontal, a diferencia de sus portales web homólogos que por lo general empujan una navegación de arriba hacia abajo. Se ha conseguido cumplir estos objetivos sin comprometer la navegación de la web.

Después de utilizar, analizar y probar la aplicación web comercial se puede comprobar que se ha conseguido este objetivo, ya que cualquier usuario podría interactuar con la aplicación y solicitar cualquiera de los servicios que ofrece con facilidad.

El desarrollo del portal web comercial ha sido una experiencia realmente enriquecedora, me ha dado la posibilidad de trabajar con lenguajes que hasta ahora me resultaban desconocidos o que conocía muy poco, aumentando mi experiencia en el ámbito informático.

6.2.- Trabajos futuros

Durante la implementación de la base d datos se creo una tabla para albergar los tiempos que transcurren entre el paso de diapositivas en los banners, la razón era para otorgar la posibilidad de tener múltiples banners funcionando a la vez con diferentes tiempos para el paso de dispositivas. En motivos es sencillo, en la mayoría de portales web comerciales abundan los banners por ser un método sencillo y efectivo para dar la información concreta, resaltándola sobre el resto de la página web.

Una posible mejora es entregar un servidor de correo electrónico. Esta opción posibilita la creación de correo electrónico propio, lo cual da una visión más profesional del negocio frente a clientes o posibles clientes. Además otorga la posibilidad de dar el formato que se desee al correo electrónico que es enviado al realizar una consulta desde la sección contacto de la página web.

Existen actualizaciones de las librerías de jQuery que otorgan más funcionalidades que se podrían usar en toda la página web. Pero no solo hay actualizaciones de jQuery, si no también de las librerías que se apoyan en jQuery, como la librería “jquery.vslider.js” que es la que genera y hace funcionar el banner.

Se puede integrar marketing por internet SMO (Social Media Optimization) al proyecto para mejorar la calidad del servicio ofrecido al cliente proporcionando medios a través de los cuales publicitar la franquicia en las redes sociales, tales como Facebook, twittter...

En la web también se pueden crear funcionalidades para las redes sociales, introduciendo botones en el menú lateral para cada red social que se desee implementar. Es otra manera de publicitarse y una muy buena opción para este tipo de portales web.

Otro posible mejora para el proyecto seria mejorar su posicionamiento en los buscadores dando de alta la web en los propios buscadores, principalmente Google, y en los directorios, tales como yahoo, isearch y Open Directory Project.

También se puede usar la estrategia Link building, la cual es una de las estrategias del SEO, muy eficaz, que consiste en conseguir que otras páginas web enlacen a la página que nos interese para hacer que los buscadores la consideren relevante y la posicionen mejor en sus rankings. La técnica del Link building puede hacerse de manera natural, cuando otras webs nos enlazan sin previo acuerdo comentando algo que hemos hecho o dicho, o bien de manera artificial, cuando simulamos que estos enlaces se han conseguido de manera natural.



7.- ANEXOS

Anexo I: Presupuesto.

CONCEPTO	PRECIO (€)
Web	7 400 ,00 €
Análisis de la aplicación	600,00 €
Creación de la línea de diseño personalizado	2 500,00 €
Maquetación	1 000,00 €
Implementación de la aplicación	3 000,00 €
Introducción de contenidos (depende de la cantidad de contenidos a introducir)	200,00 €
Pruebas	100,00 €
Marketing por Internet : SEO (opcional)	280,00 €
Análisis palabras clave	50,00 €
Alta en buscadores	80,00 €
Alta en directorios	150,00 €
Otras tareas consultar (link building, dominios extranjeros, etc.)	0,00 €
Marketing por Internet : SMO (no incluye diseño personalizado) (opcional)	200,00 €
Blog (con diseño de plantilla)	50,00 €
Página de empresa en Facebook	50,00 €
Canal de videos en YouTube	50,00 €
Cuenta en Twitter	50,00 €
Otros servicios	284,95 €
Registro del dominio 1 año	9,95 €
Alojamiento web 1 año	
• Espacio web: 1000 MB	
• Transferencia: 3 GB	
• Cuentas de correo: 50	75,00 €
• Antivirus/Antispam, PHP, MySQL	
• Subdominios ilimitados	
Instalación de la aplicación en el servidor definitivo	200,00 €
Alta en registro de accesos de Google Analytics y en el directorio de Google Local	0,00 €
Subtotal	8164,95 €
Total	8164,95 €

El presupuesto actual asciende a la cantidad de OCHO MIL CIENTO SESENTA Y CUATRO EUROS CON NOVENTA Y CINCO CÉNTIMOS (8164,95 €).

Este tipo de presupuesto contempla la gran mayoría de los aspectos más importantes a la hora de la creación de un portal web comercial, no solo de la creación de la aplicación web si no también de otros aspectos importantes como por ejemplo el posicionamiento en buscadores. Se tratan algunos de los temas planteados como posibles mejoras, ya que en este proyecto de fin de carrera se quería desarrollar un portal web que cumpla todos los objetivos planteados con el menor coste posible.

El precio total asciende del presupuesto estimado, normalmente ocurre esto, ya que es muy difícil estimar un coste para este tipo de proyectos.

Anexo II: Instalación

Para poder visualizar el portal web comercial a través de internet es necesario alojarlo en un servidor web que cumpla los siguientes requisitos:

- Comprar un dominio el domino para el portal web
- Tener un servidor web contratado o propio
- Tener instalado en el servidor web apache 2.2.21 o posterior.
- Tener instalado en el servidor web PHP en su versión 5.3.9. En caso de instalar una versión posterior o anterior de PHP no se asegura un correcto funcionamiento del portal web, debido a las diferencias entre distintas versiones de PHP.
- Tener instalado MySQL en el servidor en su versión 5.5.20.
- Se recomienda el uso de phpmyadmin para el manejo de la base de datos. No es obligatorio.

Si el servidor cumple los requisitos puede ser instalado en el servidor. Para ello es necesario:

- Crear un directorio en el servidor para albergar el portal web comercial.
- Copiar los archivos de la aplicación web al directorio creado en el servidor para tal función.
- Crear la base datos e importar el archivo BBDD.sql entregado.
- En el archivo func.php cambiar los datos de la base de datos en la líneas 328 y 329 tal y como se indica a continuación:

```
$conex= mysql_connect("Directorio de la base de datos, normalmente es  
"localhost"", "Nombre de usuario", "contraseña") or die("no se puede conectar porque  
".mysql_error());  
mysql_select_db("nombre de la base de datos");
```

En caso de querer instalar la aplicación web en un servidor local se han de cumplir los requisitos especificados anteriormente y seguir las instrucciones de instalación, pero además es necesario abrir el puerto 25 (SMTP) del router para poder enviar correos desde la sección contacto. Si el puerto esta cerrado no se podrá enviar correos y por lo tanto la funcionalidad del portal web comercial no será del 100%.

Anexo III: Manual de usuario

El portal web comercial consta de dos partes bien diferenciadas a las que los usuarios pueden acceder; pagina web principal accesible por todos los usuarios y el gestor de contenidos al que solo pueden acceder los usuarios dados de alta en el sistema. Por lo tanto, y como viene siendo habitual en esta memoria del proyecto, se van a explicar ambas partes por separado.

Pagina principal

La página web principal posee navegación lateral, por lo que para poder ver el contenido hay que desplazarse lateralmente o usar el menú de la parte izquierda del explorador. En caso de querer desplazarse manualmente por el contenido el usuario ha de usar la barra de “Scroll” inferior del explorador con que esta visualizando la pagina web.

La manera mas cómoda y sencilla para navegar por la web es a través del menú situada en la parte izquierda del explorador. Contiene todas las secciones de la página web excepto el banner publicitario. Para ver una sección de la pagina web mediante el uso del menú solo hace falta clicar en el nombre de la sección que aparece en el menú y el contenido de la pagina web se desplazara hasta llegar a la sección deseada.

El banner tiene un tiempo para el paso de diapositivas de manera automática. Se pueden desplazar las imagines de manera manual posicionando el ratón encima del banner. Cuando el ratón este encima del banner aparecerán dos flechas, una a cada lado del banner. Pulsando la flecha situada a la izquierda del banner el usuario pasa a la diapositiva anterior, si pulsa la flecha situada en la parte derecha del banner se pasa a la diapositiva siguiente.

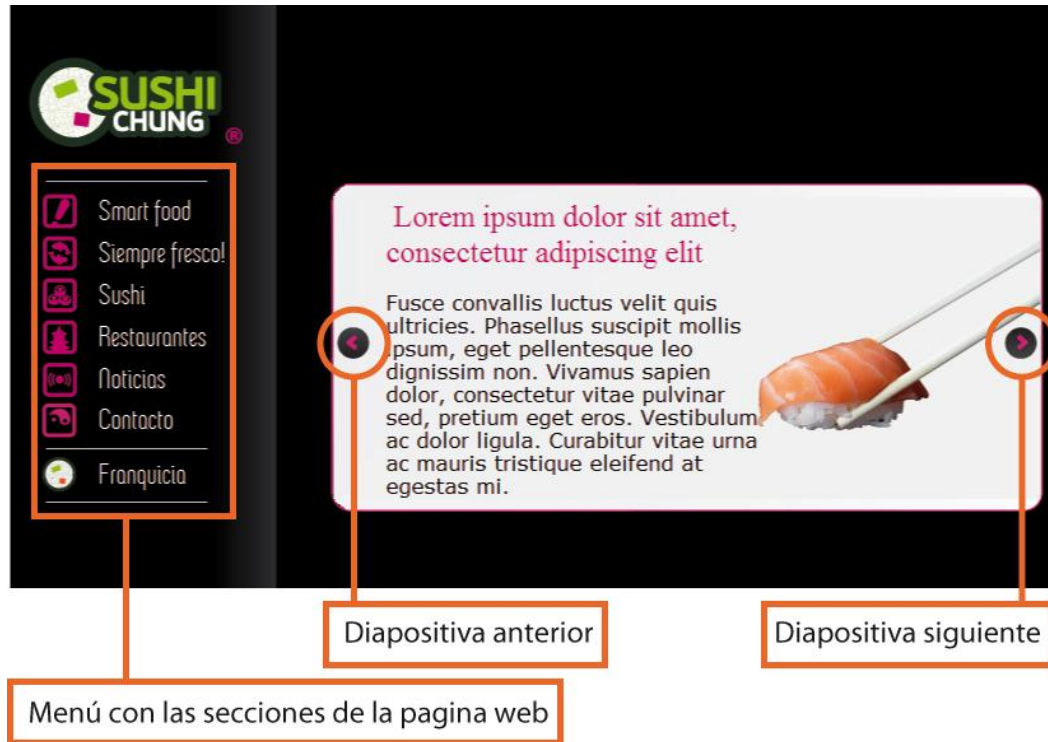
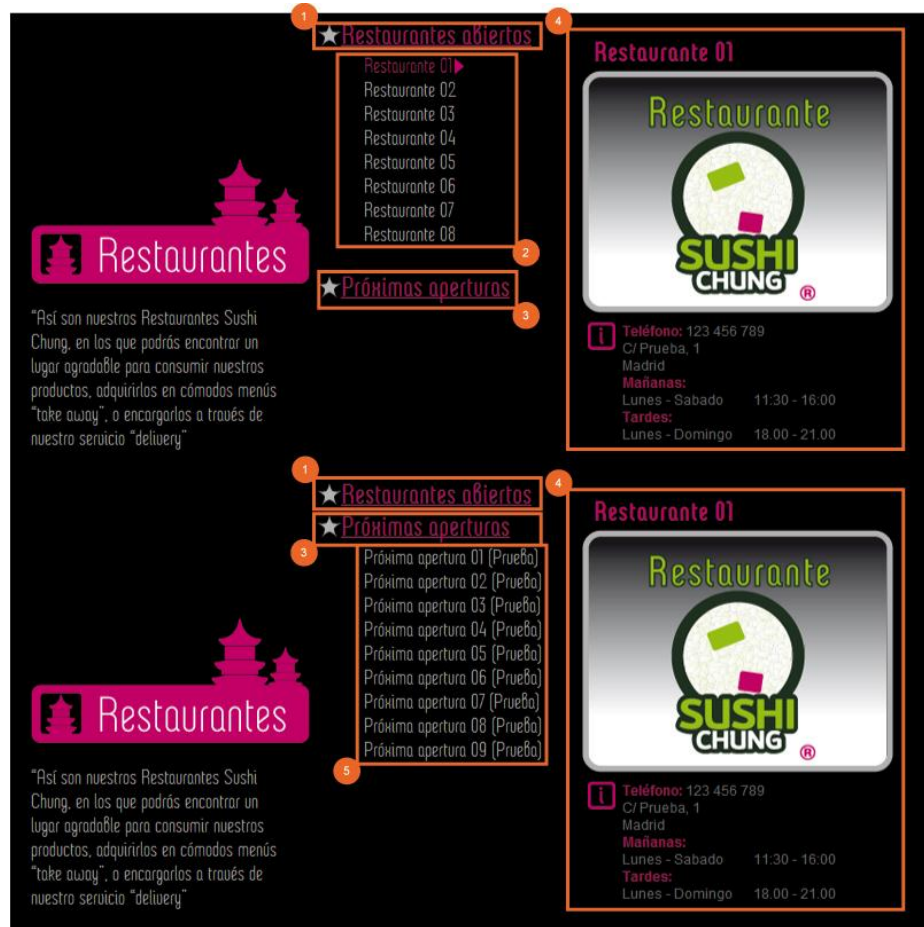


FIGURA 59 Explicación menú y banner

Las secciones Smart Food, Siempre Fresco! y Sushi contienen información de la franquicia. Para visualizarla basta con dirigirse a la sección.

La sección restaurantes consta de dos menús, uno encima del otro. Por defecto de muestra el menú de restaurante abiertos. Si el usuario quiere ver el menú denominado próximas aperturas deberá pinchar sobre el mismo. Se contraerá el menú de restaurantes abiertos y se desplegará el menú de próximas aperturas. Para volver a visualizar el menú de restaurantes abiertos tan solo hay que pinchar en el nombre del índice.

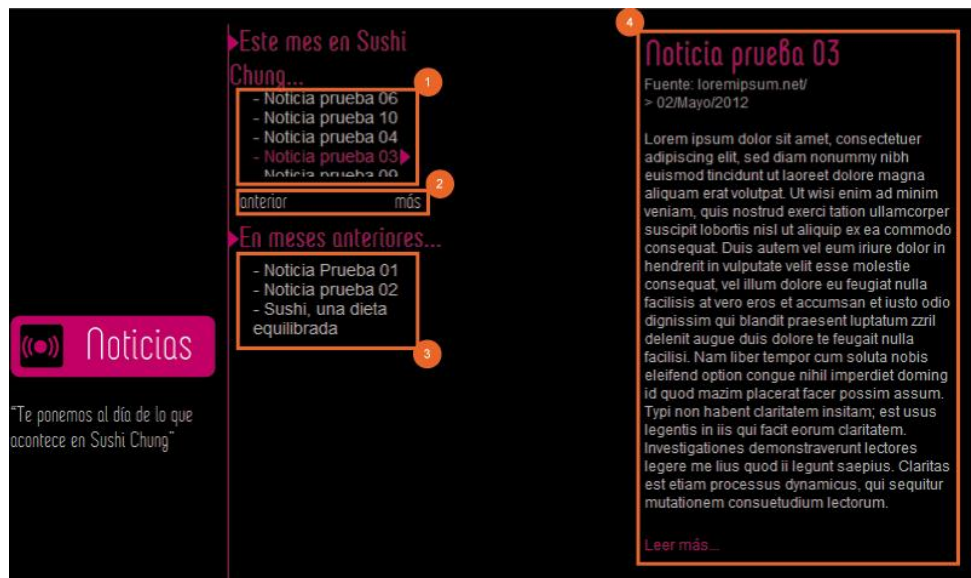
En caso de que el usuario se encuentre en el menú de restaurantes abiertos puede visualizar todos los restaurantes que se encuentren en el menú. Para ello basta con clicar en el nombre del restaurante que aparece en este menú.



- 1 Enlace cabecera del menú de restaurantes abiertos. Despliega el menú.
- 2 Menú de restaurantes abiertos.
- 3 Enlace cabecera del menú de las próximas aperturas. Despliega el menú.
- 4 Espacio donde se muestra la información de los restaurantes.
- 5 Menú de próximas aperturas.

FIGURA 60 Explicación sección restaurantes

En la sección noticias encontramos un menú dividido en dos índices. El primer índice, el que esta más arriba, contiene el índice de las noticias publicadas en el mes actual. El índice que esta mas abajo contiene las noticias de meses pasados. Para ver el contenido de una noticia el usuario tiene que clicar en el nombre de la misma que aparece en el menú de las noticias, tanto si es una noticia del mes actual como si es una noticia de meses pasado.



- 1 Menú con las noticias del mes actual.
- 2 Enlaces mas y menos para mover el contenido del menu.
- 3 Menú con las noticias de mes anteriores.
- 4 Espacio donde se muestra la información de las noticias.

FIGURA 61 Explicación sección noticias

La sección contacto contiene un formulario en donde el usuario puede enviar un mail de consulta a la franquicia, siempre que rellene correctamente los campos que contiene. En el campo nombre el usuario tiene que escribir su nombre. El campo mail tiene que ser rellenado con un mail valido, en caso de no serlo no se enviara el mail. Por ultimo ha de rellenar el campo comentarios. Ninguno de los campos de este formulario puede estar vacío, en caso contrario no será enviada la consulta.

En la sección franquicia se encuentran los datos para contactar con el departamento de expansión de franquicias. Consta de un Dossier en donde el posible futuro franquiciado vera las ventajas de unirse a la franquicia.

Gestor

Para acceder al gestor el usuario ha de ir a la url de la pagina web y añadir al final “/adm”.

Una vez el usuario este en la pagina de inicio del gestor deberá logarse con mail y contraseña con el que fue dado de alta en el sistema.

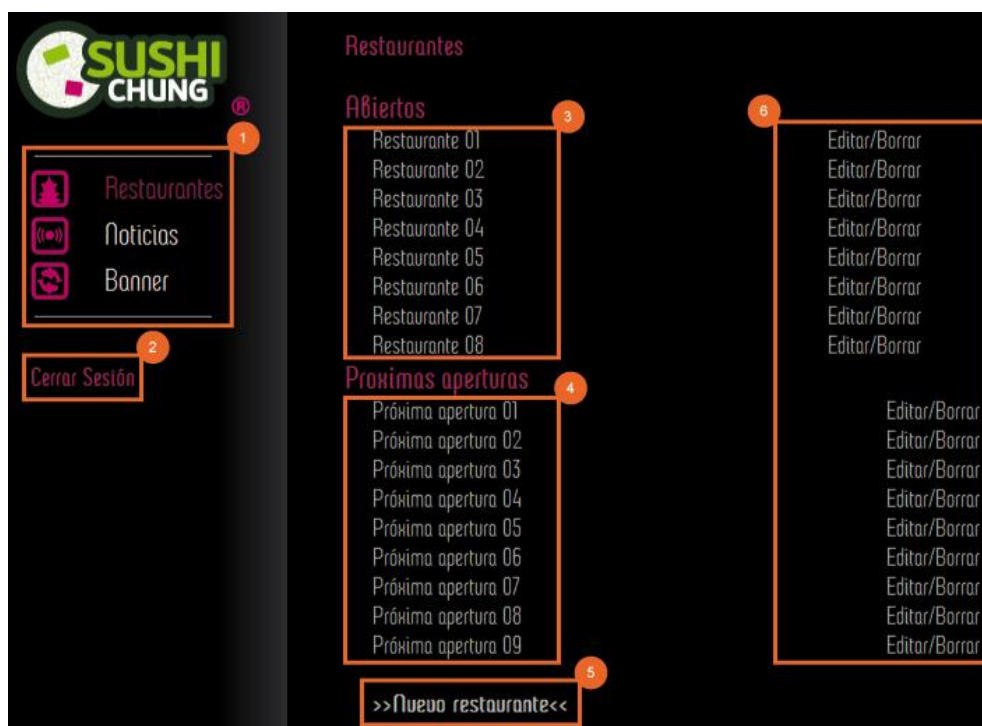
El gestor consta de un menú situado a la izquierda del explorador, el cual esta formado por todas las secciones gestionables.

Al pulsar en alguna de estas secciones se redirige al usuario a la pagina de gestión de la sección seleccionada. Todos los gestores contienen un listado con los nombres de todos los registros que las componen. Cada registro o entrada de la sección tiene a la derecha una columna de acciones en donde se presentan las acciones de edición y borrado. Si el usuario quiere editar o borrar alguna de las entradas de la sección ha de pulsar en la acción que corresponda.

Debajo de cada listado nos encontramos con un enlace para la creación de una nueva entrada. Si el usuario quiere crear una entrada de la sección ha de pulsar dicho enlace.

Tanto en la creación como en la edición se presente un formulario que debe rellenarse para la creación, o modificar algún dato del formulario para modificarlo. Existen pautas tanto para la creación como para edición de las entradas de las secciones.

En la sección restaurantes existe dos tipos de entradas, restaurante abierto y próxima apertura, esta ultima se refiere a los restaurante que aun no han sido abierto serán abierto próximamente.



- 1 Menú con las secciones del gestor
- 2 Enlace para cerrar sesión.
- 3 Menú de los restaurantes abiertos.
- 4 Menú de las próximas aperturas.
- 5 Enlace para crear un nuevo restaurante.
- 6 Enlaces para editar o borrar un restaurante.

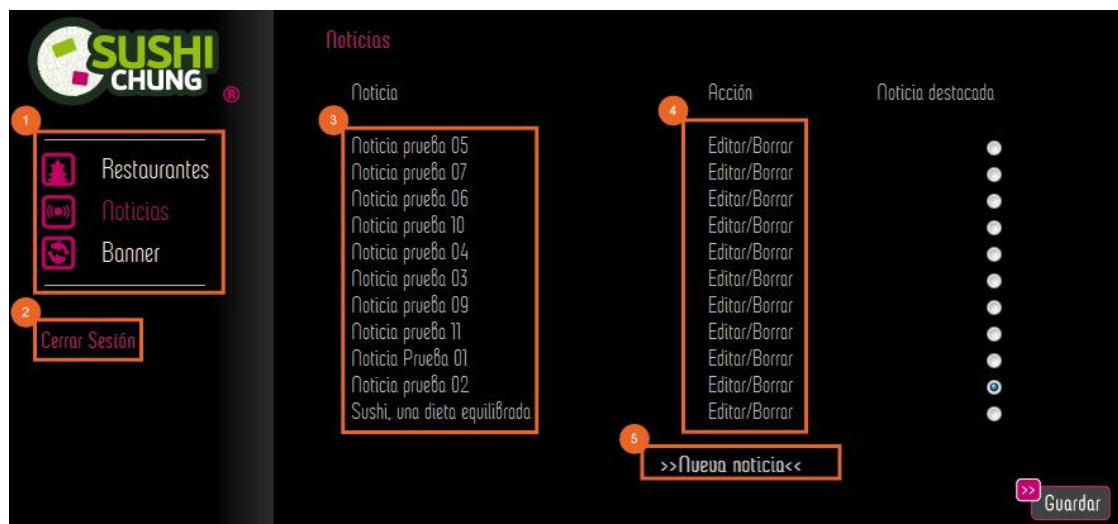
FIGURA 62 Explicación sección restaurantes del gestor

Los campos a rellenar para la creación y modificación dependen del tipo de restaurante que sea. Los campos son:

- Nombre: Campo obligatorio para todos los restaurantes.
- Foto: Campo no obligatorio independientemente del tipo de restaurante que sea. Para poder subir una foto es necesario que el archivo sea una imagen con la extensión JPG, PNG o GIF.
- Dirección: campo obligatorio para los restaurantes abiertos.
- Ciudad: Campo obligatorio para todos los tipos de restaurantes

- Teléfono: Campo obligatorio para los restaurantes abiertos.
- Legal apertura: No es obligatorio par ningún tipo de restaurante. Índice si se desea índice en la información del restaurante que los horarios del restaurante están sujetos al horario y días de apertura del centro comercial en el que se encuentran, en caso de encontrarse en un centro comercial.
- Orden: No es obligatorio par ningún tipo de restaurante, es de tipo numérico. Indica el orden de aparición del restaurante en el índice de restaurantes de la página principal. Cuando mayor es el numero antes aparece en el la lista de restaurantes de la pagina web principal. El primer restaurante que aparece en la lista es el restaurante que se muestra al crear la página.
- Horario de mañana y horario de tarde: Estos campos no son obligatorios para las próximas aperturas. Tienen dos botones debajo, botón más y botón menos. Cabe la posibilidad de que un restaurante tenga diferentes horarios dependiendo de los días de la semana. Para introducir varios horarios basta con pulsar el botón mas. Al pulsarlo aparece otro horario más. Se pueden crear hasta un máximo de 7 horarios de mañana y 7 horarios de tarde diferentes.
- Próxima apertura: Este campo sirve para indicar si el restaurante es un restaurante de próxima apertura. Si no esta activo el restaurante será tomado como un restaurante abierto.

En la sección noticias nos encontramos que la página de gestión aparece otra columna en la tabla que sirve de índice para sección. Esta columna es para indicar cual es la noticia destacada, la cual será la notica mostrada al carga la página web. Si se desea cambiar la notica destacada basta con pulsar el botón de la columna Noticia destacada de la fila en donde se encuentra la notica que se quiere pasar a notica destacada, y después parecerá un botón de guardar el que hay que pulsar para guardar el cambio.



- 1 Menú con las secciones del gestor
- 2 Enlace para cerrar sesión.
- 3 Menú de las noticias.
- 4 Enlaces para editar o borrar una noticia.
- 5 Enlace para crear una nueva noticia.
- 6 Columna donde seleccionar la noticia destacada.
- 7 Botón para guardar la selección de la noticia destacada.

FIGURA 63 Explicación sección noticias del gestor

No se puede eliminar la noticia destacada. Si se quiere eliminar la noticia destacada habrá que indicar primero cual va a ser la próxima noticia destacada, guardar los resultados y borrar la noticia que se quería borrar.

Al crear o modificar alguna noticia aparecerán los siguientes campos:

- Titular: Campo obligatorio.
- Fuente: Campo obligatorio.
- Fecha: Campo obligatorio. Pulsar en la imagen con forma de calendario de la derecha del campo.
- Contenido: Campo obligatorio.
- Foto: No es obligatorio. Si se desea subir una foto el archivo tiene que ser una imagen con extensión JPG, PNG o GIF.

- Noticia destacada: No es obligatorio. Al marcarlo se indica al sistema que es la próxima noticia destacada.
- Link: No es obligatorio.

En la página de gestión del banner en la parte inferior del contenido se encuentra el intervalo de tiempo. Dicho intervalo de tiempo es el tiempo que transcurre en el paso de diapositivas del banner de la página principal. Para modificarlo el usuario debe introducir el nuevo intervalo y pulsar el botón guardar para guardar los resultados.



- 1 Menú con las secciones del gestor
- 2 Enlace para cerrar sesión.
- 3 Menú de las diapositivas del banner.
- 4 Enlaces para editar o borrar una diapositiva.
- 5 Enlace para crear una nueva diapositiva.
- 6 Intervalo de tiempo entre el paso de diapositivas del banner en la página web.
- 7 Botón para guardar el intervalo de tiempo entre el paso de diapositivas.

FIGURA 64 Explicación sección banner del gestor

En la creación y edición de diapositivas del banner se encuentran los siguientes campos:

- Contenido: Campo obligatorio. Es el texto que aparece en la diapositiva.
- Posición del contenido: El contenido de texto de las diapositivas puede aparecer en la mitad izquierda o derecha de la diapositiva.
- Fondo: El campo es obligatorio. Es el fondo de la diapositiva. El archivo tiene que ser una imagen con extensión JPG, PNG o GIF.



8.- BIBLIOGRAFÍA

8.1.- Bibliografía web

A lo largo del proyecto, muchas páginas Web me han servido de referencia para la implementación de la aplicación Web realizada.

Las más interesantes son las siguientes:

- [1] 3/04/12 Búsqueda de información <http://www.google.com/>
- [2] 10/04/12 Definiciones <http://es.wikipedia.org>
- [3] 12/04/12 Información MySQL
<http://es.scribd.com/doc/63764883/33/Ventajas-y-Desventajas-de-MySQL>
- [4] 5/05/12 Información de putos de función
<http://www.qsm.com/resources/function-point-languages-table>
- [5] 25/03/12 Información y librerías de jQuery <http://jquery.com/>
- [6] 25/03/12 Información de funciones de PHP <http://php.net/>
- [7] 21/05/12 Información de MySQL <http://www.mysql.com/>
- [8] 10/05/12 Apuntes de Cocomo II <http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/teo/8/cocomo2-apuntes.pdf>
- [9] 6/05/12 Información sobre los diferentes lenguajes de programación web
<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>
- [10] 18/05/12 Envío de e-mail por atreves de correo SMTP
<http://blog.techwheels.net/send-email-from-localhost-wamp-server-using-sendmail/>
- [11] 23/04/12 Información, documentación y librería jQueryUi
<http://jqueryui.com/>
- [12] 04/04/12 Información, documentación y librería JQuery ScrollTo <http://flesler.blogspot.com.es/2007/10/jqueryscrollto.html>

8.2.- Bibliografía

Ha habido libros que han me han servidor de referencia durante el desarrollo del proyecto de fin de carrera, aquí se presentan los más utilizados:

- Soledad Delgado Sanz, Jorge Aurelio Tejedor Cerbel, Abraham Gutiérrez Rodríguez, Jesús Bobadilla Sancho. *JavaScript*. Editorial McGraw-Hill, 2001
- Rainer Kolbeck. *Gran Libro JavaScript*. Editorial Marcombo, 2000
- Bill Sanders. *Flash 5 avanzado*. Editorial Anaya, 1999
- Lázaro Issi Camy. *ActionScript para FLASH MX*. Editorial Anaya, 2003
- César Pérez. Macromedia Dreamweaver MX Desarrollo de aplicaciones y base de datos en la Web. Editorial RA-MA, 2003
- Germán Galeano Gil, Pablo Díaz Márquez, José Carlos Sánchez Alonso. *Manual Imprescindible de HTML 4*. Editorial Anaya, 2001
- Grady Booch, James Rumbaugh, Ivar Jacobson. *El Lenguaje Unificado de Modelado UML 2.0*. Addison-Wesley.
- Olivier Heurtel. *PHP 5.3 Desarrollar un sitio Web dinámico e interactivo*. Eni Ediciones.
- López Quijado, José. *Domine PHP y MYSQL. 2ª Edición*. Ra-Ma Editorial.
- Chaffer y Karl Swedberg *Aprende jQuery 1.3* Anaya Multimedia.